МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
КАРАЧАЕВО-ЧЕРКЕССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ У.Д. АЛИЕВА

КОМПЬЮТЕРНЫЕ СЕТИ И ИНТЕРНЕТ

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Карачаевск – 2019

УДК-004.43
ББК 32.973.26-018.1

Печатается по решению редакционно-издательского совета Карачаево-Черкесского государственного университета им. У.Д. Алиева

Лепшокова А.Н., Эльканова А.А. Компьютерные сети и интернет. Учебно-методическое пособие / А.Н. Лепшокова, Эльканова А.А. – Карачаевск: КЧГУ, 2019. – 196 с.

Составители: Эльканова *А.А.,* к. п. н.,
Лепшокова А.Н., к. п. н.

ISBN 978-5-8307-0601-8

Содержит теоретический материал и методические методические рекомендации к выполнению лабораторных работ и по решению задач на тему «Компьютерные сети и интернет» Предназначено для студентов, обучающихся по всем профилям подготовки бакалавров, изучающих следующие дисциплине: «Интернет-программирование», «Web-программирование», «Основы интернет-программирования», «Вычислительные сети, системы и мультимедиа технологии», «Компьютерные сети и системы», «Информационные компьютерные технологии».Пособие также может быть полезно для учащихся и абитуриентов для сдачи ЕГЭ.

Рецензенты: *Х.Дж. Шунгаров*, к. ф.-м. н., доцент,
Х.А.Гербеков, к. ф.-м. н., доцент.

Карачаево-Черкесский государственный университет, 2019
Эльканова А.А., Лепшокова А.Н., 2019

Введение

Всемирная паутина (World Wide Web) спровоцировала революцию в информатике, предоставив любому пользователю возможность публикации HTML-документов. До недавнего времени информация в этих документах была в большинстве случаев статической, что требовало реакции сервера на действия пользователя. С введением динамического HTML парадигма Web сместилась от взаимодействия с сервером в сторону создания интерактивных web-узлов и web-приложений. Поскольку динамический HTML обеспечивает возможность взаимодействия HTML-документов с пользователем и полного их изменения на клиентском компьютере, вы можете создавать web-приложения с богатыми возможностями.

В результате изучения данного пособия, проводимого под руководством преподавателя, студенты познакомятся с:

- технологиями и концепцией создания статических web-страниц;
- принципами разработки структуры web-страниц;
- основными элементами языка;
- средствами оформления страниц;
- способами передачи данных от удаленного пользователя на сервер;
- с адресацией и статической маршруизацией в IP-сетях;
- взаимосвязью языков скриптов и таблицей стилей для оформления web-документов;
- организацией проверки данных, введенных пользователем.

По окончании данного курса студенты смогут:

- создавать несложные статические сайты и наполнять их форматированным и структурированным содержимым;
- использовать стилевое форматирование для оформления HTML-документов;
- иметь представление о языке создания сценариев (то есть уметь понимать конструкции языка и интерпретировать результат);
- создавать web-документы с динамически изменяемым содержимым;
- использовать стилевое форматирование совместно с языками сценариев для расширения возможностей оформления документов;
- решать задачи на кодировку адреса указанного файла в сети интернет;
- решать задачи на восстановите IP-адреса;
- решать задачи на расчет количества подсетей и хостов в подсети на основе IP-адреса и маски подсети.

Практические и лабораторные занятия для студента являются одним из основных методов приобретения и углубления знаний, познания общественной практики.

Главной задачей этих занятий является развитие общих и профессиональных компетенций, умений приобретать научные знания путем личных поисков, формирование активного интереса и вкуса к творческому самостоятельному подходу в учебной, практической и лабораторной работе.

Данные занятия складываются из изучения учебной и специальной литературы как основной, так и дополнительной, нормативного материала и решения задач.

Материалы, изложенные в пособии, рассчитаны на студентов, уже имеющих представление об эталонной модели взаимодействия открытых систем и общих принципах построения локальных и глобальных сетей.

РАЗДЕЛ I. IP-АДРЕСА ИХ ПРИМЕНЕНИЕ И СТРУКТУРА

Тема 1.1. Знакомство с IP-адресами

Цели:

1. *ознакомиться с основными понятиями: IP-адрес, фаил;*
2. *ознакомиться со структурой IP-адреса;*
3. *решать задачи на кодировку адреса указанного файла в сети интернет*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации: учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М.: ИНФРА-М, 2014. - 734 с.
2. Цветкова, А.В. Информатика и информационные технологии [Электронный ресурс]: учебное пособие для СПО/ Цветкова А.В.— Электрон. текстовые данные. — Саратов: Научная книга, 2019.— 190 с.— Режим доступа: http://www.iprbookshop.ru/87074.html. — ЭБС «IPRbooks»– 376 с.
3. Яроцкая, Е.В. Географические информационные системы [Электронный ресурс]: учебное пособие/ Яроцкая Е.В., Матвеева А.В., Дьяченко А.А.— Электрон. текстовые данные. — Саратов: Ай Пи Ар Медиа, 2019.— 146 с.— Режим доступа: http://www.iprbookshop.ru/85744.html. — ЭБС «IPRbooks»

Теоретические сведения

IP-адрес (ˌaɪˈpiː-адрес; является словообразовательной полукалькой английского *IP address*, буквенной аббревиатурой от англ. *Internet Protocol Address* «адрес интернет-протокола») — уникальный сетевой адрес узла в компьютерной сети, построенной на основе стека протоколов TCP/IP.

В сети интернет требуется глобальная уникальность адреса; в случае работы в локальной сети требуется уникальность адреса в пределах сети. В версии протокола IPv4 IP-адрес имеет длину 4 байта, а в версии протокола IPv6 IP-адрес имеет длину 16 байт.

Одна часть IP-адреса представляет собой номер сети, другая – идентификатор хоста. Точно так же, как у разных домов на одной улице в адресе присутствует одно и то же название улицы, у хостов в сети в адресе имеется общий номер сети. И точно так же, как у различных домов имеется собственный номер дома, у каждого хоста в сети имеется собственный уникальный идентификационный номер – идентификатор хоста. Номер сети используется маршрутизаторами (роутерами, интернет-центрами) для передачи пакетов в нужные сети, тогда как идентификатор хоста определяет конкретное устройство в этой сети, которому должны быть доставлены пакеты.

Структура

IP-адрес состоит из четырех частей, записанных в виде десятичных чисел с точками (например, 192.168.1.1). Каждую из этих четырех частей называют октетом. Октет представляет собой восемь двоичных цифр (например, 11000000, или 192 в десятичном виде).

Таким образом, каждый октет может принимать в двоичном виде значения от 00000000 до 11111111, или от 0 до 255 в десятичном виде.

На следующем рисунке показан пример IP-адреса, в котором первые три октета (192.168.1) представляют собой номер сети, а четвертый октет (16) – идентификатор хоста.

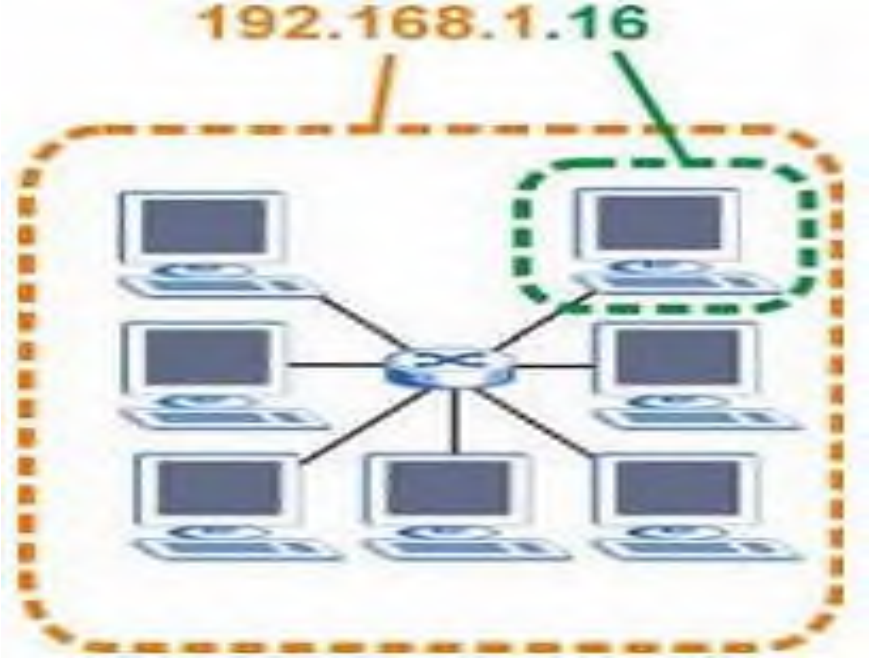


Рисунок 1. Номер сети и идентификатор хоста

Количество двоичных цифр в IP-адресе, которые приходятся на номер сети, и количество цифр в адресе, приходящееся на идентификатор хоста, могут быть различными в зависимости от маски подсети.

Частные IP-адреса

У каждого хоста в сети интернет должен быть уникальный адрес. Если ваши сети изолированы от интернета (например, связывают два филиала), для хостов без проблем можно использовать любые IP-адреса. Однако, уполномоченной организацией по распределению нумерации в сети интернет (IANA) специально для частных сетей зарезервированы следующие три блока IP-адресов:

- 10.0.0.0 — 10.255.255.255
- 172.16.0.0 — 172.31.255.255
- 192.168.0.0 — 192.168.255.255

IP-адреса указанных частный подсетей иногда называют "серыми".

IP-адреса можно получить через IANA, у своего провайдера услуг интернет или самостоятельно назначить из диапазона адресов для частных сетей.

IP-АДРЕС КОМПЬЮТЕРА

- Компьютер, находящийся в сети, имеет IP-адрес.
- Данный адрес **состоит из 4 частей, разделенных точкой** (в конце точка не ставится).
- Каждая часть IP-адреса представляет собой число **от 0 до 255**:

0..255**.** 0..255**.** 0..255**.** 0..255
например:
245.176.185.68

- IP-адрес часто представляется **в двоичной системе счисления**; тогда он представляет собой 4 байта, разделенных точками. Так, приведенный в примере адрес выглядит следующим образом:

11110101.10110000.10111001.01000100

Пример 1

Доступ к файлу **slon.txt**, находящемуся на сервере **circ.org**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .txt	Г) circ	Ж) slon
Б) ://	Д) /	
В) http	Е) .org	

Решение:

Напомним, как формируется адрес в сети интернет. Сначала указывается протокол (как правило это «ftp» или «http»), потом «://», потом сервер, затем «/», название файла указывается в конце. Таким образом, адрес будет следующим: http://circ.org/slon.txt.

Следовательно, ответ ВБГЕДЖА.

Пример 2

Доступ к файлу **tiger.doc**, находящемуся на сервере **zoo.org**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .doc	Г) ://	Ж) http
Б) zoo	Д) tiger	
В) /	Е) .org	

Решение.

Напомним, как формируется адрес в сети интернет. Сначала указывается протокол (как правило это «ftp» или «http»), потом «://», потом сервер, затем «/», название файла указывается в конце. Таким образом, адрес будет следующим: http://zoo.org/tiger.doc.

Следовательно, ответ ЖГБЕВДА .

Пример 3

Почтовый ящик **teacher** находится на сервере **shkola1.mos.ru**. В таблице фрагменты адреса электронной почты закодированы цифрами от 1 до 6. Запишите последовательность цифр, кодирующую этот адрес.

1) @	3) shkola1	5) teacher
2) .ru	4) .	6) mos

Решение.

Напомним, как формируется почтовый адрес в сети интернет. Сначала указывается имя почтового ящика, потом «@», потом сервер, на котором находится почтовый ящик. Таким образом, искомый адрес будет следующим: teacher@shkola1.mos.ru.

Ответ: 513462.

Задания для самостоятельного выполнения

Задание 1. Доступ к файлу **start.exe**, находящемуся на сервере **game.com**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) start	Г) http	Ж) ://
Б) /	Д) game	
В) .exe	Е) .com	

Задание 2. Доступ к файлу **doc.htm**, находящемуся на сервере **site.com**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) site	Г) /	Ж) http
Б) ://	Д) .htm	
В) doc	Е) .com	

Задание 3. Доступ к файлу **book.txt**, находящемуся на сервере **bibl.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) ://	Г) txt	Ж)/
Б) book.	Д) .ru	
В) bibl	Е) http	

Задание 4. Доступ к файлу **img.bmp**, находящемуся на сервере **pic.kz**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) img	Г) ://	Ж) pic.
Б) /	Д) http	
В) kz	Е) .bmp	

Задание 5. Доступ к файлу **name.gif**, находящемуся на сервере **jour.com**, осуществляется по протоколу **ftp**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .com
Б) ftp
В) jour
Г) /
Д) ://
Е) .gif
Ж) name

Задание 6. Доступ к файлу **spis.htm**, находящемуся на сервере **sch.net**, осуществляется по протоколу **ftp**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) ://
Б) spis
В) .net
Г) .htm
Д) ftp
Е) sch
Ж)/

Задание 7. Доступ к файлу **color.gif**, находящемуся на сервере **box.net**, осуществляется по протоколу **ftp**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) ftp
Б) /
В) box.
Г) color
Д) net
Е) .gif
Ж) ://

Задание 8. Доступ к файлу **test.xls**, находящемуся на сервере **school.org**, осуществляется по протоколу **ftp**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) test
Б) school
В) /
Г) ://
Д) .org
Е) .xls
Ж) ftp

Задание 9. Доступ к файлу **table.xls**, находящемуся на сервере **ofis.com**, осуществляется по протоколу **ftp**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) /
Б) ftp
В) com
Г) ://
Д) table.
Е) ofis.
Ж) xls

Задание 10. Доступ к файлу **spis.xml**, находящемуся на сервере **book.net**, осуществляется по протоколу **ftp**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .net	Г) spis	Ж) .xml
Б) /	Д) ://	
В) ftp	Е) book	

Задание 11. Доступ к файлу **math.doc**, находящемуся на сервере **obr.ru**, осуществляется по протоколу **https**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) obr	Г) .doc	Ж) https
Б) /	Д) ://	
В) .ru	Е) math	

Задание 12. Доступ к файлу **rus.doc**, находящемуся на сервере **obr.org**, осуществляется по протоколу **https**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) obr.	Г) ://	Ж) https
Б) /	Д) doc	
В) org	Е) rus.	

Задание 13. Доступ к файлу **pupil.pdf**, находящемуся на сервере **class.ru**, осуществляется по протоколу **https**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) https	Г) pupil	Ж) class
Б) .ru	Д) ://	
В) .pdf	Е) /	

Решение.

Напомним, как формируется адрес в сети интернет. Сначала указывается протокол, потом «://», потом сервер, затем «/», название файла указывается в конце. Таким образом, искомый адрес будет

следующим: **https://class.ru/pupil.pdf**. Следовательно, ответ АДЖБЕГВ.

Задание 14. Доступ к файлу **sotr.pdf**, находящемуся на сервере **org.net**, осуществляется по протоколу **https**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .pdf	Г) .net	Ж)/
Б) sotr	Д) org	
В) ://	Е) https	

Задание 15. Доступ к файлу **hello.jpg**, находящемуся на сервере **home.info**, осуществляется по протоколу **ftp**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) info	Г) /	Ж) .jpg
Б) ://	Д) hello	
В) home.	Е) ftp	

Задание 16. Доступ к файлу **com.txt**, находящемуся на сервере **mail.net**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) /	Г) .txt	Ж) com
Б) http	Д) .net	
В) ://	Е) mail	

Задание 17. Доступ к файлу **txt.net**, находящемуся на сервере **gov.org**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .net	Г) http	Ж) .org
Б) txt	Д) /	
В) ://	Е) gov	

Тема 1.2. Адресация в интернете и адрес файла на сервере

Цели:

1. *ознакомиться с основными понятиями: адрес файла, каталог;*
2. *ознакомиться с расширением файла;*
3. *решать задачи на восстановите IP-адреса*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации: учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М.: ИНФРА-М, 2014. - 734 с.
2. Цветкова А.В. Информатика и информационные технологии [Электронный ресурс]: учебное пособие для СПО/ Цветкова А.В.— Электрон. текстовые данные. — Саратов: Научная книга, 2019.— 190 с.— Режим доступа: http://www.iprbookshop.ru/87074.html. — ЭБС «IPRbooks»– 376 с.
3. Яроцкая Е.В. Географические информационные системы [Электронный ресурс]: учебное пособие/ Яроцкая Е.В., Матвеева А.В., Дьяченко А.А.— Электрон. текстовые данные. — Саратов: Ай Пи Ар Медиа, 2019.— 146 с.— Режим доступа: http://www.iprbookshop.ru/85744.html.— ЭБС «IPRbooks»

Теоретические сведения

Файл — это область в памяти компьютера (на диске), которая имеет имя и расширение, записанных через точку:

имя.расширение

Имя файлу дает его создатель, т.е. человек, который создал этот файл в определенной программе, либо сама программа задает его автоматически.

- **Расширение файла** определяет, к какому типу файлов он относится или, другими словами, какая информация в нем хранится: например, текстовый файл, файл-изображение, файл электронной таблицы, видео-файл и т.п.
- Рассмотрим некоторые распространённые расширения файлов:

Расширения текстовых файлов:
.txt .doc .odt .docx .pdf и другие

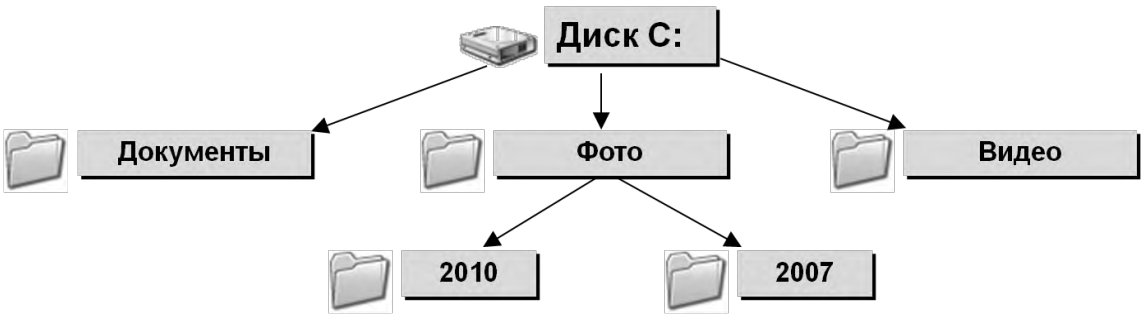
Расширения исполняемых файлов:
.com .exe и другие

Расширения файлов изображений:
.bmp .jpg .gif .png и другие

Расширения файлов веб-страниц (сайтов) и др. форматов для браузеров:
.html .html .xml .xsl и другие

Рассмотрим хранение файлов на локальном компьютере:

- файлы на диске хранятся в **каталогах** (папках);
- внутри каталогов могут находиться другие каталоги, таким образом, организуется сложная структура каталогов — иерархическая структура, называемая **деревом каталогов**;
- главный каталог диска называется **корневым каталогом** и обозначается буквой логического диска, за которой следует двоеточие и знак «\» (обратный слэш); например, **A:** – это обозначение корневого каталога диска А.



Пример дерева каталогов

- Если в примере добавить файл с именем *«file.jpg»* и расположить его в папке *2007*, то **адрес файла на диске** (или полный путь файла) будет:

C:\Фото\2007\file.jpg - полный путь файла **file.jpg**

- каждый каталог (кроме корневого) имеет один единственный **«родительский» каталог** – это тот каталог, внутри которого и располагается данный каталог; например, для каталога 2007 родительской папкой является Фото;

• **полный адрес каталога** – это перечисление всех каталогов, в которые нужно войти, чтобы попасть в данный каталог (начиная с корневого каталога диска); например:

C:\USER\BIN\SCHOOL - полный путь каталога **SCHOOL**

• таким образом, полный адрес файла состоит из адреса каталога, в котором он находится, символа \ и имени файла; еще пример:



Полный путь файла

Адрес файла в интернете состоит из перечисленных ниже составных частей:

протокол передачи данных; может быть:

- **http** (для web-страниц);
- **ftp** (для передачи файлов);
- встречается также защищенный протокол **https**;
- символы-разделители **://**, отделяющие название протокола от остальной части адреса;
- доменное имя сайта (или IP-адрес);
- может присутствовать также: каталог на сервере, где располагается файл;
- имя файла.

Каталоги на сервере разделяются прямым слэшем «/»

Пример:



Где:

1. имя протокола сетевой службы – определяет тип сервера *HTTP* (протокол передачи гипертекста);
2. разделитель в виде символа двоеточия и двух символов *Slash*;
3. полное доменное имя сервера;
4. путь поиска web-документа на компьютере;
5. имя web-сервера;
6. домен верхнего уровня *«org»*;
7. имя национального домена *«ru»*;

8. каталог *main* на компьютере;
9. каталог *news* в каталоге *main*;
10. конечная цель поиска – файл *main_news.html*.

Пример 1.

На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г:

2.17	**16**	**.65**	**8.121**
А	Б	В	Г

Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

Решение.

IP-адрес представляет собой четыре разделённых точками числа, каждое из которых не больше 255.

Под буквой В указано «.65». Так как числа в IP-адресе не могут быть больше 255, нельзя добавить в конце этого числа еще один разряд. Следовательно, этот фрагмент — последний.

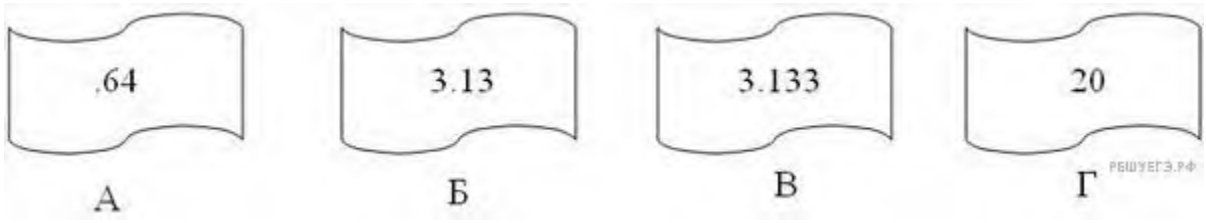
Рассмотрим фрагмент под буквой Б. Так как числа в IP-адресе не могут быть больше 255, фрагмент Б должен быть на первом месте.

В конце фрагмента Г — число 121, отделенное точкой. Так как в IP-адресе не может быть числа, большего 255, то за фрагментом Г должен следовать фрагмент, начинающийся с точки. Значит, фрагмент Г идет перед фрагментом В.

Следовательно, ответ БАГВ.

Пример 2.

Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP─адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



Решение.

IP-адрес представляет собой числа, разъединенные точками, причем числа эти не больше 255.

Посмотрим внимательнее на данные фрагменты: под буквой А мы видим «.64». Число, на которое указывает этот фрагмент, начинается с 64. Так как числа в IP-адресе не могут быть больше 255, мы не можем добавить в конце этого числа еще один разряд, а фрагментов, начинающихся с точки, больше нет, следовательно, этот фрагмент – последний.

Посмотрим на фрагмент под буквой Г. В нем стоит число без точек, значит, это либо последний фрагмент, либо первый. Место последнего фрагмента уже занято, значит, фрагмент Г на первом месте.

В конце фрагмента В - число 133, отделенное точкой. Так как в IP-адресе не может быть числа, большего 255, то за фрагментом В должен следовать фрагмент, начинающийся с точки. Значит, фрагмент В идет перед фрагментом А.

Следовательно, ответ ГБВА.

Задания для самостоятельного выполнения

Задание 1. На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г:

4.243	**116.2**	**13**	**.23**
А	Б	В	Г

Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

Задание 2. На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г:

2.12	**22**	**.30**	**5.121**
А	Б	В	Г

Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

Задание 3. Миша записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Мишина мама случайно постирала куртку вместе с запиской. После стирки Миша обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г:

17	**.44**	**4.144**	**9.13**
А	Б	В	Г

Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

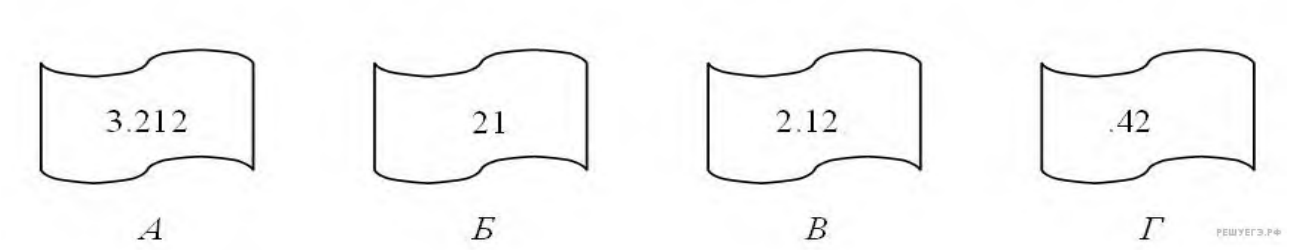
Задание 4. Костя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Костина мама случайно постирала куртку вместе с запиской. После стирки Костя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г:

.33	**3.232**	**3.20**	**23**
А	Б	В	Г

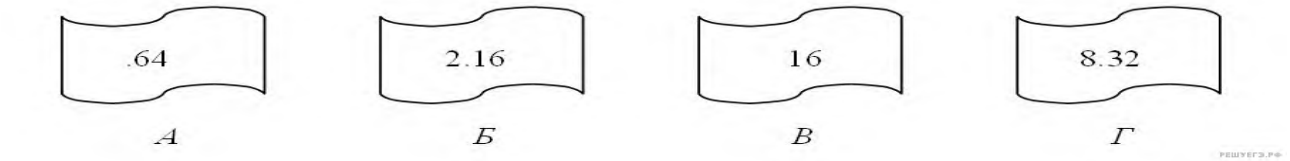
Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

Задание 5. Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены

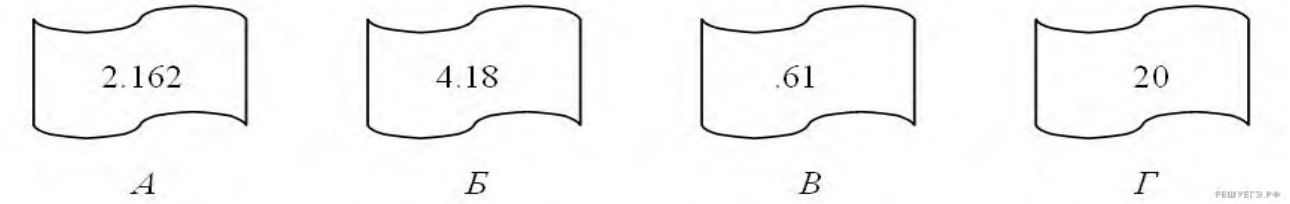
буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



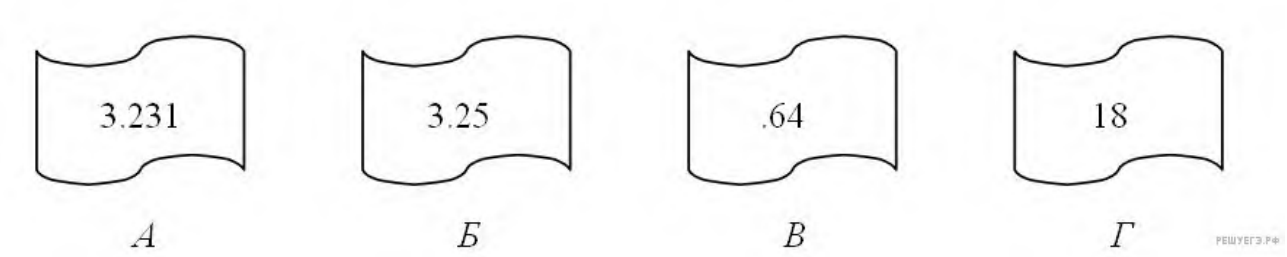
Задание 6. На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



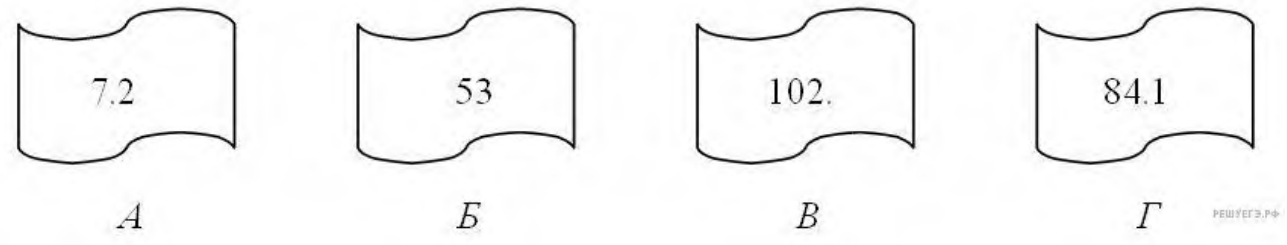
Задание 7. Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



Задание 8. Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



Задание 9. На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу. Если будет несколько вариантов решения, запишите их все через запятую.



Задание 10. На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу. Если будет несколько вариантов решения, запишите их все через запятую.

Тема 1.3. Поиск информации в интернете

Цели:

1. *ознакомиться с основными понятиями: поисковая система, поисковый сервер;*
2. *ознакомиться с тремя способами поиска в интернете;*
3. *ознакомиться с языками запросов поисковой системы;*
4. *решать задачи на поиск информации в сети интернет.*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л.

П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Цветкова, А.В. Информатика и информационные технологии [Электронный ресурс]: учебное пособие для СПО/ Цветкова А.В.— Электрон. текстовые данные.— Саратов: Научная книга, 2019.— 190 с.— Режим доступа: http://www.iprbookshop.ru/87074.html.— ЭБС «IPRbooks»– 376 с.
3. Яроцкая, Е.В. Географические информационные системы [Электронный ресурс]: учебное пособие/ Яроцкая Е.В., Матвеева А.В., Дьяченко А.А.— Электрон. текстовые данные.— Саратов: Ай Пи Ар Медиа, 2019.— 146 с.— Режим доступа: http://www.iprbookshop.ru/85744.html.— ЭБС «IPRbooks»

Теоретические сведения

Сеть интернет растет очень быстрыми темпами, поэтому найти нужную информацию среди сотен миллиардов web-страниц и сотен миллионов файлов становится все сложнее. Для поиска информации используются специальные поисковые системы, которые содержат постоянно обновляемую информацию о местонахождении web-страниц и файлов на сотнях миллионов серверов интернета.

Поисковые системы содержат тематически сгруппированную информацию об информационных ресурсах Всемирной паутины в базах данных. Специальные программы-роботы периодически "обходят" web-серверы интернета, читают все встречающиеся документы, выделяют в них ключевые слова и заносят в базу данных интернет-адреса документов.

Большинство поисковых систем разрешают автору web-сайта самому внести информацию в базу данных, заполнив регистрационную анкету. В процессе заполнения анкеты разработчик сайта вносит адрес сайта, его название, краткое описание содержания сайта, а также ключевые слова, по которым легче всего будет найти сайт.

Поиск по ключевым словам. Поиск документа в базе данных поисковой системы осуществляется с помощью введения запросов в **поле поиска**.

Запрос должен содержать одно или несколько ключевых слов, которые являются главными для этого документа. Например, для поиска самих систем поиска в интернете можно в поле поиска ввести ключевые слова "российская система поиска информации интернет"

Через некоторое время после отправки запроса поисковая система вернет список интернет-адресов документов, в которых были найдены заданные ключевые слова. Для просмотра этого документа в браузере достаточно активизировать указывающую на него ссылку.

Если ключевые слова были выбраны неудачно, то список адресов документов может быть слишком большим (может содержать десятки и даже сотни тысяч ссылок). Для того чтобы уменьшить список, можно в поле поиска ввести дополнительные ключевые слова или воспользоваться каталогом поисковой системы.

Одной из наиболее полных и мощных поисковых систем является Google (www.google.ru), в базе данных которой хранятся 8 миллиардов web-страниц и каждый месяц программы-роботы заносят в нее 5 миллионов новых страниц. В Рунете (российской части интернета) обширные базы данных, содержащие по 200 миллионов документов, имеют поисковые системы Яndex (www.yandex.ru) и Rambler (www.rambler.ru).

Поиск в иерархической системе каталогов. В базе данных поисковой системы web-сайты группируются в **иерархические тематические каталоги**, которые являются аналогами тематического каталога в библиотеке.

Тематические разделы верхнего уровня, например: *интернет, компьютеры, наука и образование* и т. д., содержат вложенные каталоги. Например, каталог интернет может содержать подкаталоги *поиск, почта* и др.

Поиск информации в каталоге сводится к выбору определенного каталога, после чего пользователю будет представлен список ссылок на интернет-адреса наиболее посещаемых и содержательных web-сайтов. Каждая ссылка обычно аннотирована, т. е. содержит короткий комментарий к содержанию документа.

Наиболее полный многоуровневый иерархический тематический каталог русскоязычных интернет-ресурсов имеет поисковая система Апорт (www.aport.ru). Каталог содержит подробную аннотацию содержания web-сайтов и указание на их географическое положение.

Поиск файлов. Для поиска файлов на серверах файловых архивов существуют специализированные поисковые системы, в том числе

поисковая система FileSearch (www.filesearch.ru). Для поиска файла необходимо ввести имя файла в поле поиска, и поисковая система выдаст интернет-адреса серверов файловых архивов, на которых хранится файл с заданным именем.

Поиск информации в русскоязычной части интернета с помощью наиболее поисковых систем: Google, Rambler, Апорт, Япс1ех и файловой поисковой системы Research можно производить с использованием интегрированной поисковой системы Gogle.ru. Для этого достаточно ввести ключевые слова в строку поиска, с помощью переключателей установить тип необходимой информации и щелкнуть по кнопке с названием поисковой системы.

ТРИ СПОСОБА ПОИСКА В ИНТЕРНЕТЕ

Интернет в целом и Всемирная паутина, в частности, предоставляют абоненту доступ к тысячам серверов и миллионам web-страниц, на которых хранится невообразимый объем информации. Как не потеряться в этом "информационном океане"? Для этого необходимо научиться искать и находить нужную информацию в сети.

Как уже было сказано, существуют три основных способа поиска информации в интернете.

1. *Указание адреса страницы.* Это самый быстрый способ поиска, но его можно использовать только в том случае, если точно известен адрес документа.

2. *Передвижение по гиперссылкам.* Это наименее удобный способ, так как с его помощью можно искать документы, только близкие по смыслу текущему документу. Если текущий документ посвящен, например, музыке, то, используя гиперссылки этого документа, вряд ли можно будет попасть на сайт, посвященный спорту.

3. *Обращение к поисковому серверу (поисковой системе)*. Использование поисковых серверов - наиболее удобный способ поиска информации. В настоящее время в русскоязычной части Интернета популярны следующие поисковые серверы:

Yandex;
Rambler;
Апорт.

Существуют и другие поисковые системы. Например, эффективная система поиска реализована на сервере почтовой службы mail.ru.

ПОИСКОВЫЕ СЕРВЕРЫ

Наиболее доступным и удобным способом поиска информации во Всемирной паутине является использование поисковых систем. При этом поиск информации можно осуществлять по каталогам, а также по набору ключевых слов, характеризующих отыскиваемый текстовый документ.

Рассмотрим использование поисковых серверов более подробно. **Поисковый сервер** содержит большое количество ссылок на самые различные документы, и все эти ссылки систематизированы в тематические каталоги. Например: спорт, кино, автомобили, игры, наука и др. Причем эти ссылки устанавливаются сервером самостоятельно, в автоматическом режиме путем регулярного просмотра всех появляющихся во Всемирной паутине web-страниц. Кроме того, поисковые серверы предоставляют пользователю возможность поиска информации по ключевым словам. После ввода ключевых слов поисковый сервер начинает просматривать документы на других web-серверах и выводить на экран ссылки на те документы, в которых встретились указанные слова. Обычно результаты поиска сортируются по убыванию специального рейтинга документов, который показывает, насколько полно заданный документ отвечает условиям поиска или насколько часто он запрашивается в сети.

ЯЗЫК ЗАПРОСОВ ПОИСКОВОЙ СИСТЕМЫ

Группа ключевых слов, сформированная по определенным правилам - с помощью языка запросов, называется запросом к поисковому серверу. Языки запросов к разным поисковым серверам очень похожи. Подробнее об этом можно узнать, посетив раздел "Помощь" нужного поискового сервера. Рассмотрим правила формирования запросов на примере поисковой системы Яndex.

Синтаксис оператора	**Что означает оператор**	**Пример запроса**
пробел или &	Логическое И (в пределах предложения)	лечебная физкультура
&&	Логическое И (в пределах документа)	рецепты && (плавленый сыр)
|	Логическое ИЛИ	фото | фотография | снимок | фотоизображение

+	Обязательное наличие слова в найденном документе	+быть или +не быть
( )	Группирование слов	(технология | изготовление) (сыра | творога)
~	Бинарный оператор И НЕ (в пределах предложения)	банки ~ закон
~~ или _	Бинарный оператор И НЕ (в пределах документа)	путеводитель по Парижу ~~ (агентство | тур)
/(n m)	Расстояние в словах (минус (-) - назад, плюс (+) - вперед)	поставщики /2 кофе музыкальное /(-2 4) образование вакансии ~ /+1 студентов
" "	Поиск фразы	"красная шапочка" Эквивалентно: красная /+1 шапочка
&&/(n m)	Расстояние в предложениях (минус (-) - назад, плюс (+) - вперед)	банк && /1 налоги

Чтобы получить лучшие результаты поиска, необходимо запомнить несколько простых правил:

1. Не искать информацию только по одному ключевому слову.

2. Лучше не вводить ключевые слова с прописной буквы, так как это может привести к тому, что не будут найдены те же слова, написанные со строчной буквы.

3. Если в итоге поиска вы не получили никаких результатов, проверьте, нет ли в ключевых словах орфографических ошибок.

Современные поисковые системы предоставляют возможность подключения к сформированному запросу семантического анализатора. С его помощью можно, введя какое-либо слово, выбрать документы, в которых встречаются производные от этого слова в различных падежах, временах и пр.

ОСУЩЕСТВЛЕНИЕ ПОИСКА ИНФОРМАЦИИ В ИНТЕРНЕТЕ

Задачи данного типа призваны проверить умение осуществлять поиск информации в интернете. Как правило от студента требуется расставить количество найденных поисковой системой страниц в порядке убывания или возрастания. Обычно начинаются такие задачи со слов: «В таблице приведены запросы к поисковому серверу…»

Пример 1

В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдет поисковый сервер по каждому запросу. Для обозначения логической операции ИЛИ в запросе используется символ |, а для логической операции И – символ &.

1	Франция | Испания | История
2	Франция & Карта & История
3	Франция | История
4	Франция & История

Решение:

В запросах у нас встречается 4 разных слова — Франция, Испания, История и Карта. Для решения мы будем использовать очень наглядный инструмент — круги Эйлера. Итак, каждое слово из запросов мы представим в виде круга.



При этом, если у нас два слова в запросе объединяются логической операцией ИЛИ (кстати, эту операцию называют дизъюнкция), то в результате получается новая фигура, объединяющая два исходных. Давайте посмотрим это на примере:

Возьмем такой запрос — Франция | Испания и построим его графический эквивалент



Полученная фигура закрашена желтым цветом. Сразу видно, что ее площадь больше, чем площадь отдельных запросов Франция или Испания. Делаем вывод, что **объединение слов в запросе логической операцией ИЛИ увеличивает количество найденных страниц**.

Теперь рассмотрим, что произойдет, если слова объединить логической операцией И (эту операцию называют конъюнкция)



Как мы видим, в данном случае полученная фигура (она выделена желтым цветом) имеет площадь меньше, чем площади исходных слов. Значит, **количество страниц, найденных по запросу с использованием логической операции И будет меньше, чем для отдельных слов**.

Вот теперь мы готовы к окончательному решению задачи. Для этого построим графический эквивалент для каждого запроса. Мы получим 4 фигуры, потом нам нужно будет просто сравнить их площади, а затем расставить их в требуемом порядке. Желтым цветом выделены фигуры, соответствующие запросу.

1) Франция | Испания | История



2) Франция & Карта & История



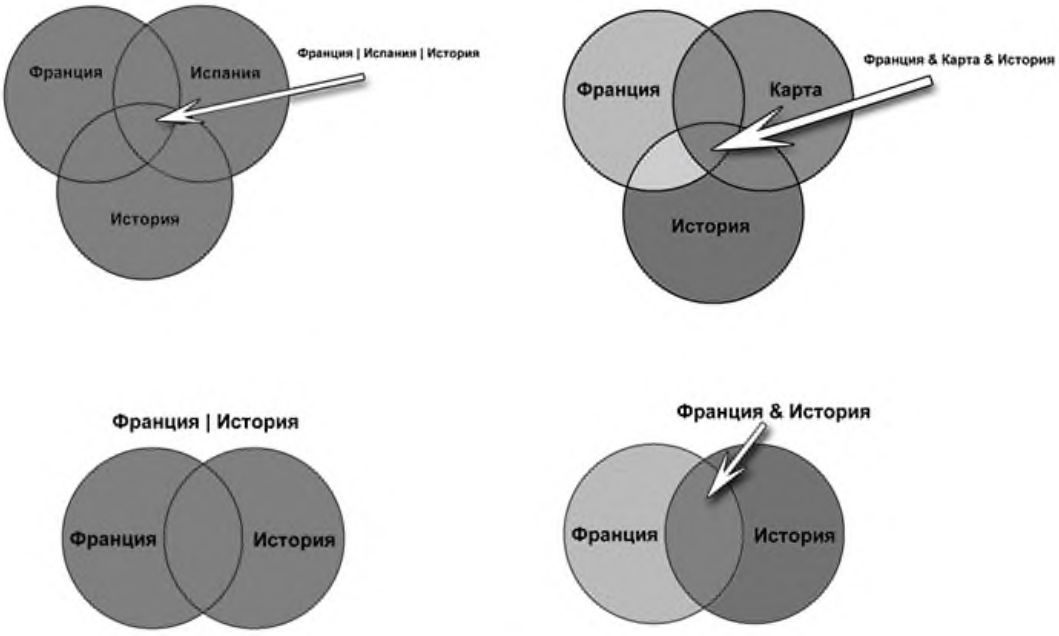
3) Франция | История



4) Франция & История



Если теперь посмотреть на полученные фигуры и сравнить площади желтых участков, то довольно просто расположить их в порядке возрастания:



Франция & Карта & История
Франция & История
Франция | История
Франция | Испания | История.
А значит **правильный ответ** будет 2431

Пример 2.

В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По

всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Лебедь | Рак | Щука
Б	Лебедь & Рак
В	Лебедь & Рак & Щука
Г	Лебедь | Рак

Решение.

Чем больше в запросе операций «ИЛИ», тем больше результатов выдаёт поисковой сервер. Чем больше в запросе операций «И», тем меньше результатов выдаст поисковой сервер. Таким образом, ответ ВБГА.

Пример 3.

В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Солнце | Воздух | Вода
Б	(Солнце & Воздух) | Вода
В	Солнце & Воздух
Г	Солнце & Воздух & Вода

Решение.

Чем больше в запросе «ИЛИ», тем больше результатов выдаёт поисковой сервер. Чем больше в запросе операций «И», тем меньше результатов выдаст поисковой сервер. Таким образом, ответ ГВБА.

Задания для самостоятельного выполнения

Задание 1. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г.

Расположите коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Солнце | Воздух | Вода
Б	Солнце & Воздух
В	Солнце & Воздух & Вода
Г	Солнце | Воздух

Задание 2. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Лебедь | Рак | Щука
Б	(Лебедь & Рак) | Щука
В	Лебедь & Рак
Г	Лебедь & Рак & Щука

Задание 3. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Рыжий | Честный | Влюблённый | Лис

Б	Рыжий | Честный | Влюблённый
В	Рыжий & Честный
Г	Рыжий | Честный

Задание 4. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Эльфы | Гномы | Орки
Б	Эльфы & Гномы & Орки
В	(Эльфы | Гномы) & Орки
Г	Эльфы | Гномы

Задание 5. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	(Муха & Денежка) | Самовар
Б	Муха & Денежка & Базар & Самовар
В	Муха | Денежка | Самовар
Г	Муха & Денежка & Самовар

Задание 6. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания

количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	(Толстой | рассказы) & повести
Б	Толстой & повести
В	Чехов & повести & Толстой & рассказы
Г	Толстой | рассказы | повести

Задание 7. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	рассказы | повести | Толстой | Чехов
Б	рассказы | повести
В	рассказы & повести
Г	рассказы | повести | Толстой

Задание 8. В таблице приведены запросы к поисковому серверу. Расположите коды запросов в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	рассказы & Толстой & Чехов & повести
Б	(рассказы | Толстой) & (Чехов | повести)
В	(рассказы | Толстой) & Чехов
Г	Толстой | рассказы | повести

Задание 9. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	повести & рассказы & Толстой
Б	(повести & рассказы) | Толстой
В	повести & рассказы
Г	Чехов & повести & рассказы & Толстой

Задание 10. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	мопсы & (паспорт | родословная)
Б	мопсы & родословная
В	(мопсы | пудели) & (паспорт | родословная)
Г	мопсы & пудели & родословная & паспорт

Задание 11. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	мопсы
Б	(мопсы & паспорт) & (родословная | пудели)
В	мопсы & паспорт
Г	мопсы | пудели

Задание 12. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	мопсы & пудели & паспорт & родословная
Б	пудели & (родословная | паспорт)
В	(пудели | мопсы) & (родословная | паспорт)
Г	мопсы & пудели & (родословная | паспорт)

Задание 13. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов в порядке **возрастания** количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&».

Код	Запрос
А	А. Конан Дойль & Г. Бичер-Стоу & Джером К. Джером
Б	А. Конан Дойль | Г. Бичер-Стоу | Джером К. Джером
В	А. Конан Дойль | (Г. Бичер-Стоу & Джером К. Джером)
Г	Г. Бичер-Стоу & Джером К. Джером

Тема 1.4. Подсчет количества адресов в сети

Цели:

1. *ознакомиться с основными понятиями: сеть, подсеть, маска подсети, размер сети;*
2. *ознакомиться с форматом записи, форматированием и планированием подсетей;*
3. *решать задачи на расчет количества подсетей и хостов в подсети на основе IP-адреса и маски подсети*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Цветкова А.В. Информатика и информационные технологии [Электронный ресурс]: учебное пособие для СПО/ Цветкова А.В.— Электрон. текстовые данные.— Саратов: Научная книга, 2019.— 190 с.— Режим доступа: http://www.iprbookshop.ru/87074.html.— ЭБС «IPRbooks»– 376 с.
3. Яроцкая Е.В. Географические информационные системы [Электронный ресурс]: учебное пособие/ Яроцкая Е.В., Матвеева А.В., Дьяченко А.А.— Электрон. текстовые данные.— Саратов: Ай Пи Ар Медиа, 2019.— 146 с.— Режим доступа: http://www.iprbookshop.ru/85744.html.— ЭБС «IPRbooks»

Теоретические сведения

МАСКИ ПОДСЕТИ

Маска подсети используется для определения того, какие биты являются частью номера сети, а какие – частью идентификатора хоста (для этого применяется логическая операция конъюнкции – "И").

Маска подсети включает в себя 32 бита. Если бит в маске подсети равен "1", то соответствующий бит IP-адреса является частью номера сети. Если бит в маске подсети равен "0", то соответствующий бит IP-адреса является частью идентификатора хоста.

Таблица 1. Пример выделения номера сети и идентификатора хоста в IP-адресе

	1-ый октет: (192)	**2-ой октет: (168)**	**3-ий октет: (1)**	**4-ый октет: (2)**
IP-адрес (двоичный)	11000000	10101000	00000001	00000010
Маска подсети (двоичная)	**11111111**	**11111111**	**11111111**	00000000
Номер сети	**11000000**	**10101000**	**00000001**	
Идентификатор хоста				00000010

Маски подсети всегда состоят из серии последовательных единиц, начиная с самого левого бита маски, за которой следует серия последовательных нулей, составляющих в общей сложности 32 бита.

Маску подсети можно определить как количество бит в адресе, представляющих номер сети (количество бит со значением "1"). Например, "8-битной маской" называют маску, в которой 8 бит – единичные, а остальные 24 бита – нулевые. Маски подсети записываются в формате десятичных чисел с точками, как и IP-адреса. В следующих примерах показаны двоичная и десятичная запись 8-битной, 16-битной, 24-битной и 29-битной масок подсети.

Таблица 2. Маски подсети

	Двоичная 1-ый октет:	**Двоичная 2-ой октет:**	**Двоичная 3-ий октет:**	**Двоичная 4-ый октет:**	**Десятичная**
8-битная маска	11111111	00000000	00000000	00000000	255.0.0.0

16-битная маска	11111111	11111111	00000000	00000000	255.255.0.0
24-битная маска	11111111	11111111	11111111	00000000	255.255.255.0
29-битная маска	11111111	11111111	11111111	11111000	255.255.255.248

РАЗМЕР СЕТИ

Количество разрядов в номере сети определяет максимальное количество хостов, которые могут находиться в такой сети. Чем больше бит в номере сети, тем меньше бит остается на идентификатор хоста в адресе. IP-адрес с идентификатором хоста из всех нулей представляет собой IP-адрес сети (192.168.1.0 с 24-битной маской подсети, например). IP-адрес с идентификатором хоста из всех единиц представляет собой широковещательный адрес данной сети (192.168.1.255 с 24-битной маской подсети, например). Так как такие два IP-адреса не могут использоваться в качестве идентификаторов отдельных хостов, максимально возможное количество хостов в сети вычисляется следующим образом:

Таблица 3. Максимально возможное число хостов

Маска подсети		**Размер идентификатора хоста**		**Максимальное количество хостов**
8 бит	255.0.0.0	24 бит	$2^{24} – 2$	16777214
16 бит	255.255.0.0	16 бит	$2^{16} – 2$	65534
24 бит	255.255.255.0	8 бит	$2^{8} – 2$	254

29 бит	255.255.255.248	3 бит	$2^3 - 2$	6

ФОРМАТ ЗАПИСИ

Поскольку маска всегда является последовательностью единиц слева, дополняемой серией нулей до 32 бит, можно просто указывать количество единиц, а не записывать значение каждого октета. Обычно это записывается как "/" после адреса и количество единичных бит в маске.

Например, адрес 192.1.1.0 /25 представляет собой адрес 192.1.1.0 с маской 255.255.255.128. Некоторые возможные маски подсети в обоих форматах показаны в следующей таблице.

Таблица 4. Альтернативный формат записи маски подсети

Маска подсети	**Альтерна-тивный формат записи**	**Последний октет (в двоичном виде)**	**Последний октет (в десятичном виде)**
255.255.255.0	/24	0000 0000	0
255.255.255.128	/25	1000 0000	128
255.255.255.192	/26	1100 0000	192
255.255.255.224	/27	1110 0000	224
255.255.255.240	/28	1111 0000	240
255.255.255.248	/29	1111 1000	248
255.255.255.252	/30	1111 1100	252

ФОРМИРОВАНИЕ ПОДСЕТЕЙ

С помощью подсетей одну сеть можно разделить на несколько. В приведенном ниже примере администратор сети создает две подсети,

чтобы изолировать группу серверов от остальных устройств в целях безопасности.

В этом примере сеть компании имеет адрес 192.168.1.0. Первые три октета адреса (192.168.1) представляют собой номер сети, а оставшийся октет – идентификатор хоста, что позволяет использовать в сети максимум $2^8 – 2 = 254$ хостов.

Сеть компании до ее деления на подсети показана на следующем рисунке.

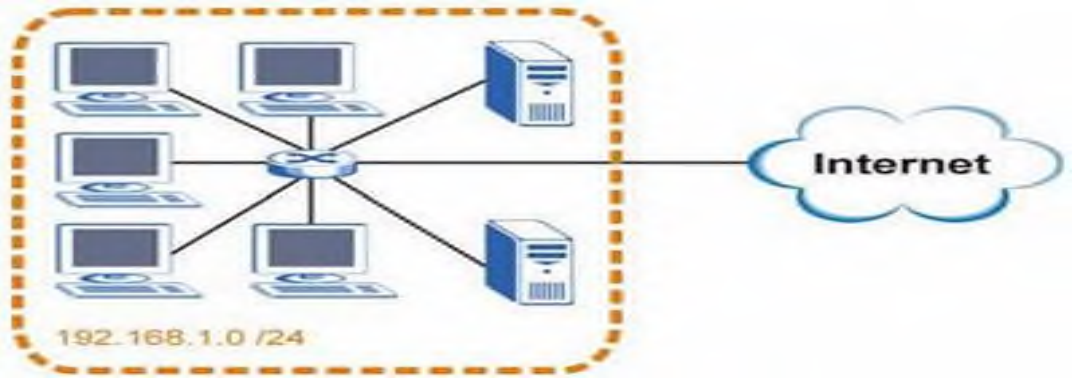
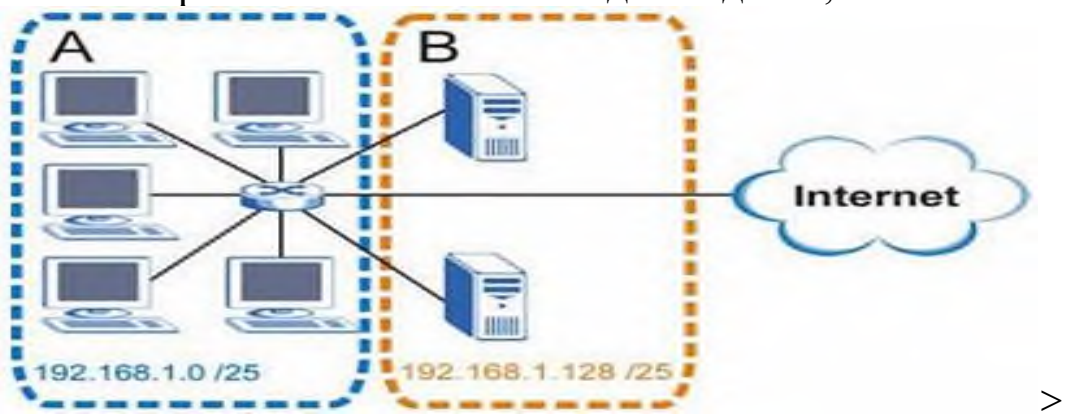


Рисунок 7. Пример формирования подсетей: до разделения на подсети

Чтобы разделить сеть 192.168.1.0 на две отдельные подсети, можно "позаимствовать" один бит из идентификатора хоста. В этом случае маска подсети станет 25-битной (255.255.255.128 или /25).

"Одолженный" бит идентификатора хоста может быть либо нулем, либо единицей, что дает нам две подсети: 192.168.1.0 /25 и 192.168.1.128 /25.

Сеть компании после ее деления на подсети показана на следующем рисунке. Теперь она включает в себя две подсети, **A** и **B**.



>

Рисунок 8. Пример формирования подсетей: после деления на подсети

В 25-битной подсети на идентификатор хоста выделяется 7 бит, поэтому в каждой подсети может быть максимум $2^7 – 2 = 126$ хостов

(идентификатор хоста из всех нулей – это сама подсеть, а из всех единиц – широковещательный адрес для подсети).

Адрес 192.168.1.0 с маской 255.255.255.128 является адресом подсети **A**, а 192.168.1.127 с маской 255.255.255.128 является ее широковещательным адресом. Таким образом, наименьший IP-адрес, который может быть закреплен за действительным хостом в подсети **A** – это 192.168.1.1, а наибольший – 192.168.1.126. Аналогичным образом диапазон идентификаторов хоста для подсети **B** составляет от 192.168.1.129 до 192.168.1.254.

Пример: четыре подсети.

В предыдущем примере было показано использование 25-битной маски подсети для разделения 24-битного адреса на две подсети. Аналогичным образом для разделения 24-битного адреса на четыре подсети потребуется "одолжить" два бита идентификатора хоста, чтобы получить четыре возможные комбинации (00, 01, 10 и 11). Маска подсети состоит из 26 бит (11111111.11111111.11111111.**11**000000), то есть 255.255.255.192.

Каждая подсеть содержит 6 битов идентификатора хоста, что в сумме дает $2^6 – 2 = 62$ хоста для каждой подсети (идентификатор хоста из всех нулей – это сама подсеть, а из всех единиц – широковещательный адрес для подсети).

Таблица 5. Подсеть 1

IP-адрес/маска подсети	**Номер сети**	**Значение последнего октета**
IP-адрес (десятичный)	192.168.1.	0
IP-адрес (двоичный)	11000000.10101000.00000001.	**00**000000
Маска подсети (двоичная)	11111111.11111111.11111111.	**11**000000
Адрес подсети 192.168.1.0	Наименьший идентификатор хоста: 192.168.1.1	

Широковещательный адрес 192.168.1.63	Наибольший идентификатор хоста: 192.168.1.62

Таблица 6. Подсеть 2

IP-адрес/маска подсети	**Номер сети**	**Значение последнего октета**
IP-адрес	192.168.1.	64
IP-адрес (двоичный)	11000000.10101000.00000001.	**01**000000
Маска подсети (двоичная)	11111111.11111111.11111111.	**11**000000
Адрес подсети 192.168.1.64	Наименьший идентификатор хоста: 192.168.1.65	
Широковещательный адрес 192.168.1.127	Наибольший идентификатор хоста: 192.168.1.126	

Таблица 7. Подсеть 3

IP-адрес/маска подсети	**Номер сети**	**Значение последнего октета**
IP-адрес	192.168.1.	128
IP-адрес (двоичный)	11000000.10101000.00000001.	**10**000000
Маска подсети (двоичная)	11111111.11111111.11111111.	**11**000000

Адрес подсети 192.168.1.128	Наименьший идентификатор хоста: 192.168.1.129
Широковещательный адрес 192.168.1.191	Наибольший идентификатор хоста: 192.168.1.190

Таблица 8. Подсеть 4

IP-адрес/маска подсети	**Номер сети**	**Значение последнего октета**
IP-адрес	192.168.1.	192
IP-адрес (двоичный)	11000000.10101000.00000001.	**11**000000
Маска подсети (двоичная)	11111111.11111111.11111111.	**11**000000
Адрес подсети 192.168.1.192	Наименьший идентификатор хоста: 192.168.1.193	
Широковещательный адрес 192.168.1.255	Наибольший идентификатор хоста: 192.168.1.254	

Пример: восемь подсетей

Аналогичным образом для создания восьми подсетей используется 27-битная маска (000, 001, 010, 011, 100, 101, 110 и 111). Значения последнего октета IP-адреса для каждой подсети показаны в следующей таблице.

Таблица 9. Восемь подсетей

Подсеть	**Адрес подсети**	**Первый адрес**	**Последний адрес**	**Широковещательный адрес**

1	0	1	30	31
2	32	33	62	63
3	64	65	94	95
4	96	97	126	127
5	128	129	158	159
6	160	161	190	191
7	192	193	222	223
8	224	225	254	255

ПЛАНИРОВАНИЕ ПОДСЕТЕЙ

Сводная информация по планированию подсетей для сети с 24-битным номером сети приводится в следующей таблице.

Таблица 10. Планирование подсетей для сети с 24-битным номером

Количество "одолженных" битов идентификатора хоста	**Маска подсети**	**Количество подсетей**	**Количество хостов в подсети**
1	255.255.255.128 (/25)	2	126
2	255.255.255.192 (/26)	4	62

3	255.255.255.224 (/27)	8	30
4	255.255.255.240 (/28)	16	14
5	255.255.255.248 (/29)	32	6
6	255.255.255.252 (/30)	64	2
7	255.255.255.254 (/31)	128	1

Пример расчета количества подсетей и хостов в подсети на основе IP-адреса и маски подсети

Приведем пример расчета количества подсетей и хостов для сети 59.124.163.151/27.

/27 - префикс сети или сетевая маска

В формате двоичных чисел 11111111 11111111 11111111 11100000

В формате десятичных чисел 255.255.255.224

В четвертом поле (последний октет) 11100000 первые 3 бита определяют число подсетей, в нашем примере $2^3 = 8$.

В четвертом поле (последний октет) 11100000 последие 5 бит определяют число хостов подсети, в нашем примере $2^5 = 32$.

Диапазон IP первой подсети 0~31 (32 хоста), но 0 - это подсеть, а 31 - это Broadcast. Таким образом, максимальное число хостов данной подсети - 30.

Первая подсеть: 59.124.163.0

Broadcast первой подсети: 59.124.163.31

Диапазон IP второй подсети с 59.124.163.32 по 59.124.163.63

Вторая подсеть: 59.124.163.32

Broadcast второй подсети: 59.124.163.63

Мы можем высчитать диапазон IP восьмой подсети с 59.124.163.224 по 59.124.163.255

Восьмая подсеть: 59.124.163.224

Broadcast восьмой подсети: 59.124.163.255

В нашем примере IP-адрес 59.124.163.151 находится в пятой подсети.
Пятая подсеть: 59.124.163.128/27
Диапазон IP пятой подсети с 59.124.163.128 по 59.124.163.159
Broadcast пятой подсети: 59.124.163.159

Пример 1.

Маской подсети называется 32-разрядное двоичное число, которое определяет, какая часть IP-адреса компьютера относится к адресу сети, а какая часть IP-адреса определяет адрес компьютера в подсети. В маске подсети старшие биты, отведенные в IP-адресе компьютера для адреса сети, имеют значение 1; младшие биты, отведенные в IP-адресе компьютера для адреса компьютера в подсети, имеют значение 0.

Если маска подсети 255.255.255.224 и IP-адрес компьютера в сети 162.198.0.157, то порядковый номер компьютера в сети равен_____

Решение.

1. Так как первые три октета (октет - число маски, содержит 8 бит) все равны 255, то в двоичном виде они записываются как 24 единицы, а значит, первые три октета определяют адрес сети.

2. Запишем число 224 в двоичном виде.

$224_{10}=11100000_2$

3. Запишем последний октет IP-адреса компьютера в сети:

$157_{10}=10011101_2$

4. Сопоставим последний октет маски и адреса компьютера в сети:

111<u>00000</u>

100<u>11101</u>

Подчеркнута нужная нам часть, отвечающая (по условию) за адрес компьютера в подсети. Переведем её в десятичную систему счисления:

$11101_2=28_{10}$.

Задания для самостоятельного выполнения

Задание 1. Маской подсети называется 32-разрядное двоичное число, которое определяет, какая часть IP-адреса компьютера относится к адресу сети, а какая часть IP-адреса определяет адрес компьютера в подсети. В маске подсети старшие биты, отведенные в IP-адресе компьютера для адреса сети, имеют значение 1; младшие

биты, отведенные в IP-адресе компьютера для адреса компьютера в подсети, имеют значение 0.

Если маска подсети 255.255.255.192 и IP-адрес компьютера в сети 10.18.134.220, то номер компьютера в сети равен_____

Задание 2. Если маска подсети 255.255.248.0 и IP-адрес компьютера в сети 112.154.133.208, то номер компьютера в сети равен_____

Задание 3. Если маска подсети 255.255.224.0 и IP-адрес компьютера в сети 206.158.124.67, то номер компьютера в сети равен_____

Задание 4. В терминологии сетей TCP/IP маской подсети называется 32-разрядное двоичное число, определяющее, какие именно разряды IP-адреса компьютера являются общими для всей подсети – в этих разрядах маски стоит 1. Обычно маски записываются в виде четверки десятичных чисел - по тем же правилам, что и IP-адреса. Для некоторой подсети используется маска 255.255.254.0. Сколько различных адресов компьютеров теоретически допускает эта маска, если два адреса (адрес сети и широковещательный) не используют?

Задание 5. Для некоторой подсети используется маска 255.255.255.192. Сколько различных адресов компьютеров теоретически допускает эта маска, если два адреса (адрес сети и широковещательный) не используют?

Задание 6. Для некоторой подсети используется маска 255.255.248.0. Сколько различных адресов компьютеров допускает эта маска?

Примечание. На практике для адресации компьютеров не используются два адреса: адрес сети и широковещательный адрес.

Задание 7. Для некоторой подсети используется маска 255.255.255.128. Сколько различных адресов компьютеров теоретически допускает эта маска, если два адреса (адрес сети и широковещательный) не используют?

Задание 8. Если маска подсети 255.255.252.0 и IP-адрес компьютера в сети 226.185.90.162, то номер компьютера в сети равен_____

Задание 9. Для некоторой подсети используется маска 255.255.255.224. Сколько различных адресов компьютеров теоретически допускает эта маска, если два адреса (адрес сети и широковещательный) не используют?

Задание 10. Если маска подсети 255.255.240.0 и IP-адрес компьютера в сети 232.126.150.18, то номер компьютера в сети равен_____

Задание 11. В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места — нули. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 98.162.71.94 адрес сети равен 98.162.71.64. Чему равно наибольшее количество возможных адресов в этой сети?

Задание 12. Для узла с IP-адресом 98.162.71.94 адрес сети равен 98.162.71.64. Чему равно наименьшее количество возможных адресов в этой сети?

Тема 1.5 Определение адреса и маски сети

Цели:

1. *ознакомиться с основными понятиями: сеть, адрес, маска и размер сети;*
2. *ознакомиться с форматом записи, форматированием и планированием подсетей;*
3. *решать задачи на расчет количества подсетей и хостов в подсети на основе IP-адреса и маски подсети*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Цветкова А.В. Информатика и информационные технологии [Электронный ресурс]: учебное пособие для СПО/ Цветкова А.В.— Электрон. текстовые данные.— Саратов: Научная книга, 2019.— 190 с.— Режим доступа: http://www.iprbookshop.ru/87074.html.— ЭБС «IPRbooks»– 376 с.
3. Яроцкая Е.В. Географические информационные системы [Электронный ресурс]: учебное пособие/ Яроцкая Е.В., Матвеева А.В., Дьяченко А.А.— Электрон. текстовые данные.— Саратов: Ай Пи Ар Медиа, 2019.— 146 с.— Режим доступа: http://www.iprbookshop.ru/85744.html.— ЭБС «IPRbooks»

Теоретические сведения

Все адреса, используемые в современных компьютерных сетях – это 32-разрядные двоичные числа.

Для вычисления адреса сети компьютер применяет к адресу узла сети и маске, представленным в двоичном виде, операцию *поразрядной конъюнкции*. Это значит, что в определенном разряде итогового числа стоит 1 тогда и только тогда, когда у обоих исходных чисел в этом разряде стоит 1. При разборе задания эта операция обозначается символом &. Таким образом,

адрес_ сети = адрес_ узла & маска.

2. Чтобы пояснить операцию поразрядной конъюнкции, разберем несколько примеров. В этих примеров для удобства двоичные разряды рассматриваемых двоичных чисел собраны в группы по 4 разряда. Такие группы удобно обозначать 16-чными цифрами.

Пример 1. Пусть X = 1101; Y = 1011. Тогда X&Y = 1001 (только в первом и последнем разряде и в X, и в Y стоит 1).

Пример 2. Пусть X = 1101; Y = 0011. Тогда X&Y = 0001

Пример 3. Пусть X = 1111 1101; Y = 1010 0011. Тогда X&Y = 1010 0001

Пример 4. Пусть X = 1111 1111; Y = 1010 0011. Тогда X&Y = 1010 0011

Для любого 8-разрядного числа Y выполнено: **11111111 & Y = Y**

Пример 5. Пусть X = 0000 0000; Y = 1010 0011. Тогда X&Y = 0000 0000

Для любого 8-разрядного числа Y выполнено: **00000000 & Y = 00000000**

3. Работать с 32-разрядными двоичными числами человеку неудобно, поэтому для IP-адресов используют побайтное представление, записывая каждый байт (т.е. восьмерку двоичных разрядов, иногда говорят - октет) в десятичной системе счисления, например IP-адрес 11011001 11101001 11101000 00000011 можно записать как 217.233.232.3. Байты (точнее - их десятичные представления) принято отделять точками.

Свойства поразрядной конъюнкции, которые отмечены в примерах 4 и 5, можно записать так. Пусть A–целое число от 0 до 255, тогда

A & 255 = A

A & 0 = 0

4. При выполнении задания полезно учитывать следующее.

1. В маске подсети старшие (левые) байты всегда 255 (т.е. в них все разряды 1), младший байт (четвертый слева, т.е. самый правый) всегда 0 (т.е. в нем все разряды 0).

2. В третьем слева байте слева стоит какое-то количество единиц и далее – нули.

Поэтому третье слева число в маске может принимать только такие значения:

. $128 = 1000\ 0000_2 = 255\text{-}127$

. $192 = 1100\ 0000_2 = 128+64 = 255 - 63$

. $224 = 1110\ 0000_2 = 128+64+32 = 255 - 31$

. $240 = 1111\ 0000_2 = 128+64+32+16 = 255 - 15$

. $248 = 1111\ 1000_2 = 128+64+32+16+8 = 255 - 7$

. $252 = 1111\ 1100_2 = 128+64+32+16+8+4 = 255 - 3$

. $254 = 1111\ 1110_2 = 128+64+32+16+8+4+2 = 255 - 1$

. $255 = 1111\ 1111_2 = 128+64+32+16+8+4+2+1$

3. Таким образом:

- первое и второе число в адресе сети те же, что и в адресе узла сети;

- четвертое – всегда 0;

- третье получается из третьего числа адресе узла сети обнулением определенного количества младших разрядов. Например, если третье число в маске подсети равно 248, то обнуляются три младших разряда третьего числа адреса узла подсети.

Пример решения задач.

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и его маске. По заданным IP-адресу узла и маске определите адрес сети:

IP-адрес: 145.92.137.88 Маска: 255.255.240.0

При записи ответа выберите из приведенных в таблице чисел 4 фрагмента четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без точек.

A	B	C	D	E	F	G	H
0	145	255	137	128	240	88	92

Пример. Пусть искомый адрес сети 192.168.128.0 и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет HBAF.

Решение:

1. Запишем числа маски сети в двоичной системе счисления.

$255_{10}=11111111_2$

$240_{10}=11110000_2$

$0_{10}=00000000_2$

2. Адрес сети получается в результате поразрядной конъюнкции чисел маски и чисел адреса узла (в двоичном коде). Так как конъюнкция 0 с чем-либо всегда равна 0, то на тех местах, где числа маски равны 0, в адресе узла стоит 0. Аналогично, там, где

числа маски равны 255, стоит само число, так как конъюнкция 1 с любым числом всегда равна этому числу.

3. Рассмотрим конъюнкцию числа 240 с числом 137.

$$240_{10}=11110000_2$$

$$137_{10}=10001001_{10}$$

Результатом конъюнкции является число $10000000_2=128_{10}$.

4.Сопоставим варианты ответа получившимся числам: 145, 92, 128, 0.

Ответ ВНЕА

Определение маски

IP-адрес (v4) состоит из 32-бит. Это можно взять в рамочку, как в школьных учебниках. Желательно запомнить и про IPv6 тоже: 128 бит.

Теоретически **IPv4-адресов** может быть: $2^{32} = 2^{10}*2^{10}*2^{10}*2^2 = 1024*1024*1024*4 \approx 1000*1000*1000*4 = 4$ млрд.

Всего 4 миллиарда. Но дальше будет рассмотрено, сколько из них не используется, грубо говоря, съедается.

Как записывается IPv4-адрес? Он состоит из четырёх октетов и записывается в десятичном представлении без начальных нулей, октеты разделяются точками: например, "192.168.11.10".

Если что, октет — это ровно то же самое, что байт. Но если вы скажете "октет" в среде профессионалов, они вас сразу зауважают и вам легче будет сойти за своего.

В заголовке IP-пакета есть поля "source IP" и "destination IP". Это адреса источника: кто посылает и назначения: кому отправлено. Почти как на почтовом конверте. Внутри пакетов у IP-адресов нет никаких масок, и разделителей между октетами тоже нет. Просто 32 бита для адреса назначения и еще 32 для адреса источника.

Однако, когда **IP-адрес** присваивается интерфейсу — ещё говорят, сетевому адаптеру — компьютера или маршрутизатора, то, кроме самого адреса этого устройства, ему присваивают еще и **маску подсети**.

Можно повторить, это важно: ***маска IP-адреса*** НЕ передается в заголовках IP-пакетов.

Компьютерам маска подсети нужна для определения границ... угадайте, чего именно... подсети. Это нужно, чтобы каждый мог определить, кто находится с ним в одной (под)сети, а кто — за ее пределами. Вообще-то можно говорить просто "сети", часто этот термин используют именно в значении "IP-подсеть". Внутри одной

сети компьютеры обмениваются пакетами напрямую, но если нужно послать пакет в другую сеть, шлют их шлюзу по умолчанию (это третий параметр, настраиваемый в сетевых свойствах). Вот как это происходит.

Маска подсети — это тоже 32-бита. Но, в отличие от **IP-адреса**, нули и единицы в ней не могут чередоваться. Всегда сначала идут единицы, потом нули.

- Не может быть маски

120.22.123.12=01111000.00010110.01111011.00001100.

- Но может быть маска

255.255.248.0=11111111.11111111.11111000.00000000.

Сначала N единиц, потом 32-N нулей. Легко догадаться, что такая форма записи избыточна. Вполне хватило бы числа N, называемого длиной маски. Так и делают: пишут 192.168.11.10/21 вместо 192.168.11.10 255.255.248.0. Обе формы имеют один и тот же смысл, но первая заметно удобнее.

Чтобы определить границы подсети, компьютер делает побитовое умножение (логическое И) между **IP-адресом и маской**, а на выходе получает адрес с обнулёнными битами в позициях нулей маски.

Рассмотрим пример 192.168.11.10/21:

11000000.10101000.00001011.00001010

11111111.11111111.11111000.00000000

11000000.10101000.00001000.00000000 = 192.168.8.0

Класс A, количество битов, заимствованных из адресной части хоста	Маска подсети	Число возможных подсетей	Число хостов/подсеть	Число битов в маске подсети
1	255.128.0.0	2	8388606	/9
2	255.192.0.0	4	4194302	/10
3	255.224.0.0	8	2097150	/11
4	255.240.0.0	16	1048574	/12
5	255.248.0.0	32	524286	/13
6	255.252.0.0	64	262142	/14
7	255.254.0.0	128	131070	/15
8	255.255.0.0	256	65534	/16
9	255.255.128.0	512	32766	/17
10	255.255.192.0	1024	16382	/18
11	255.255.224.0	2048	8190	/19
12	255.255.240.0	4096	4094	/20
13	255.255.248.0	8192	2046	/21
14	255.255.252.0	16384	1022	/22
15	255.255.254.0	32768	510	/23
16	255.255.255.0	65536	254	/24
17	255.255.255.128	131072	126	/25
18	255.255.255.192	262144	62	/26
19	255.255.255.224	524288	30	/27
20	255.255.255.240	1048576	14	/28
21	255.255.255.248	2097152	6	/29
22	255.255.255.252	4194304	2	/30
23	255.255.255.254	8388608	2*	/31

Пример решения задач.

В терминологии сетей TCP/IP маска сети — это двоичное число, меньшее 2^{32}; в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места нули. Маска определяет, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес — в виде четырёх байт, причём каждый байт записывается в виде

десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32. 240.0.

Для узла с IP-адресом 224.128.112.142 адрес сети равен 224.128.64.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Решение.

Рассмотрим третий слева байт в IP-адресе узла и адресе сети, представим их в двоичном виде:

$112_{10} = 0111\ 0000_2$; $64_{10} = 0100\ 0000_2$.

Маской сети является такое двоичное число, которое при поразрядной конъюнкции с IP-адресом узла даст адрес сети, при этом первая часть числа состоит из единиц, а всё остальное − нули. Таким числом является $1100\ 0000_2 = 192_{10}$.

Ответ: 192.

Задания для самостоятельного выполнения

Задание 1. В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу IP-адресу узла и его маске. По заданным IP-адресу сети и маске определите адрес сети:

IP-адрес: 146.212.200.55 Маска: 255.255.240.0

При записи ответа выберите из приведенных в таблице чисел 4 фрагмента четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без точек.

A	B	C	D	E	F	G	H
0	212	146	240	200	192	55	255

Пример. Пусть искомый адрес сети 192.168.128.0 и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет HBAF.

Задание 2. По заданным IP-адресу узла и маске определите адрес сети.

IP адрес узла: 217.9.142.131

Маска: 255.255.192.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	9	16	64	128	142	192	217

Задание 3. По заданным IP-адресу узла и маске определите адрес сети.

IP –адрес узла: 217.9.142.131

Маска: 255.255.224.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	9	64	128	131	142	192	217

Задание 4. По заданным IP-адресу узла и маске определите адрес сети.

IP –адрес узла: 142.9.199.145

Маска: 255.255.192.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	9	16	64	128	142	192	224

Задание 5. По заданным IP-адресу узла и маске определите адрес сети.

IP –адрес узла: 142.9.227.146

Маска: 255.255.224.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H

0	9	16	64	128	142	192	224

Задание 6. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 217.9.191.133

Маска: 255.255.192.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	9	16	64	128	142	192	217

Задание 7. По заданным IP-адресу узла и маске определите адрес сети.

IP –адрес узла: 217.8.162.162

Маска: 255.255.224.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	8	16	64	128	160	162	217

Задание 8. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 217.19.128.131

Маска: 255.255.192.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	16	19	64	128	131	192	217

Задание 9. По заданным IP-адресу узла и маске определите адрес сети.

IP –адрес узла: 224.9.195.133

Маска: 255.255.192.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	9	16	128	133	192	195	224

Задание 10. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 224.230.250.29

Маска: 255.255.240.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	19	208	224	230	240	248	255

Задание 11. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 224.120.249.18

Маска: 255.255.240.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	19	120	208	224	240	248	255

Задание 12. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 224.31.249.137

Маска: 255.255.240.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H

255	249	240	224	137	31	8	0

Задание 13. В терминологии сетей TCP/IP маска сети — это двоичное число, меньшее 2^{32}; в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места нули. Маска определяет, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес — в виде четырёх байт, причём каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32. 240.0.

Для узла с IP-адресом 224.128.114.142 адрес сети равен 224.128.64.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Задание 14. Для узла с IP-адресом 224.128.114.142 адрес сети равен 224.128.96.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Задание 15. Для узла с IP-адресом 117.191.88.37 адрес сети равен 117.191.80.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Задание 16. В терминологии сетей TCP/IP маской сети называется 32-разрядная двоичная последовательность, определяющая, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. Обычно маска записывается по тем же правилам, что и IP-адрес – в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа.

Пример. Пусть IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0. Тогда адрес сети равен 231.32.240.0.

Для узла с IP-адресом 235.116.177.140 адрес сети равен 235.116.160.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Задание 17. Для узла с IP-адресом 227.138.127.144 адрес сети равен 227.138.64.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Задание 18. Для узла с IP-адресом 111.81.208.27 адрес сети равен 111.81.192.0. Чему равно наименьшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Задание 19. Для узла с IP-адресом 111.81.208.27 адрес сети равен 111.81.192.0. Чему равно наибольшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Задание 20. Для узла с IP-адресом 111.81.200.27 адрес сети равен 111.81.192.0. Чему равно наибольшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Задание 21. Для узла с IP-адресом 111.81.208.27 адрес сети равен 111.81.192.0. Чему равно наименьшее возможное значение второго слева байта маски? Ответ запишите в виде десятичного числа.

Задание 22. Для узла с IP-адресом 111.81.200.27 адрес сети равен 111.81.192.0. Чему равно наименьшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Задание 23. Для узла с IP-адресом 119.167.50.77 адрес сети равен 119.167.48.0. Чему равно наименьшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Задание 24. Идентификатор некоторого ресурса сети интернет имеет следующий вид:

http://www.ftp.ru/index.html

Какая часть этого идентификатора указывает на протокол, используемый для передачи ресурса? Выпишите нужную часть.

Задание 25. Доступ к файлу ftp.net , находящемуся на сервере txt.org, осуществляется по протоколу http. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А	Б	В	Г	Д	Е	Ж
.net	ftp	://	http	/	.org	txt

Задание 26. На сервере test.edu находится файл demo.net, доступ к которому осуществляется по протоколу http. Фрагменты адреса данного файла закодированы буквами А, Б ... Ж (см. таблицу). Запишите последовательность этих букв, которая кодирует адрес указанного файла в Интернете.

А	Б	В	Г	Д	Е	Ж
test	demo	://	/	http	.edu	.net

Задание 27. На сервере info.edu находится файл exam.net, доступ к которому осуществляется по протоколу http. Фрагменты адреса данного файла закодированы буквами a, Ь, c ... g (см. таблицу). Запишите последовательность этих букв, которая кодирует адрес указанного файла в Интернете.

a	b	c	d	e	f	g
info	/	.net	.edu	http	exam	://

Задание 28. Доступ к файлу index.html, размещенному на сервере www.ftp.ru, осуществляется по протоколу http. В таблице приведены фрагменты адреса этого файла, обозначенные буквами от А до З. Запишите последовательность этих букв, соответствующую адресу данного ф

айла.

А	Б	В	Г	Д	Е	Ж	З
.html	www.	/	ftp	.ru	http	index	://

Раздел II. HTML (HyperText Markup Language)

Тема 2.1. Структура HTML-документа

Цели:

1. *ознакомиться с основными понятиями: элемент, атрибут;*
2. *ознакомиться со структурой HTML-документа;*
3. *создать простейшую HTML-страничку;*
4. *научиться использовать комментарии.*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

HTML содержит типы элементов, представляющих параграфы, гипертекстовые ссылки, списки, таблицы, изображения и т.д. Каждое объявление типа элемента обычно описывает три части: начальный тег, содержимое и конечный тег.

Название элемента появляется в начальном теге (<название-элемента>) и в конечном теге (</название-элемента>). Некоторые элементы HTML допускают отсутствие конечного тега. Некоторые

типы элементов HTML не имеют содержимого. Такие пустые элементы никогда не имеют конечного тега. Названия элементов всегда нечувствительны к регистру. Элементы могут иметь ассоциированные свойства, называемые атрибутами, которые могут иметь значения (по умолчанию или устанавливаемые автором). Пары атрибут/значение появляются перед конечным символом ">" начального тега элемента. Любое количество (допустимое) пар значений атрибута, разделённых пробелами, может появляться в начальном теге элемента. Они могут появляться в любом порядке. По умолчанию требуется, чтобы все значения атрибутов были ограничены с использованием двойных или одинарных кавычек, однако в некоторых случаях можно устанавливать значение атрибута без использования кавычек, но рекомендуется использовать знак кавычек даже тогда, когда можно обойтись без него. Названия атрибутов всегда нечувствительны к регистру.

Комментарии HTML имеют следующий синтаксис:

<!-- это комментарий -->
<!-- и это тоже комментарий,
занимающий более одной строки -->

Элемент HTML

Описание: определяет начало и конец HTML-документ.

Начальный тег: ***не обязателен***

Конечный тег: ***не обязателен***

Пример:

<HTML>
...элементы head, body и т.п. идут здесь...
</HTML>

Элемент HEAD

Описание: содержит информацию о текущем документе, такую как название, ключевые слова и другие данные, не являющиеся содержимым документа.

Начальный тег: ***не обязателен***

Конечный тег: ***не обязателен***

Элемент TITLE

*Описание**:*** определяет заголовок страницы или название окна. Каждый документ HTML **обязан** содержать элемент TITLE в разделе HEAD. Авторы должны использовать элемент TITLE для идентификации содержимого документа. Поскольку пользователи часто обращаются к документам вне контекста, авторы должны

предоставлять осмысленные заголовки. Заголовок не может содержать разметку (в том числе и комментарии).

Начальный тег: ***необходим***

Конечный тег: ***необходим***

Пример:

<HTML> <HEAD><TITLE>Колледж права и экономики</TITLE> *...другие элементы заголовка...* </HEAD> ... </HTML>	В заголовке браузера отображается надпись: Колледж права и экономики.

Метаданные.

HTML позволяет авторам специфицировать метаданные - информацию о самом документе, а не о его содержимом - различными способами.

Элемент META

Описание: Элемент **META** можно использовать для идентификации свойств документа (напр., автора, конечной даты использования, списка ключевых слов и т.д.) и установки значений этих свойств. Каждый элемент **META** определяется в разделе HEAD и определяет пару свойство-значение. Атрибут name идентифицирует свойство, а атрибут *content* определяет значение свойства. Например, следующее объявление устанавливает значение для свойства Author(автор): <META name="Author" content="Студент группы …">. Обычно META специфицирует ключевые слова, которые используются поисковыми машинами для повышения качества и скорости поиска. Например, следующее объявление устанавливает значения для свойства keywords (ключевые слова): <META name="keywords" content="Chelaybinsk, КПиЭ">

Начальный тег: ***требуется***

Конечный тег: ***запрещён***

Определения атрибутов:

name = строка – устанавливает имя свойства

content = строка – определяет значение свойства

Пример:

<HTML> <HEAD> <TITLE> Физико-математичекий факультет </TITLE>	В заголовке браузера отображается надпись: Физико-математический факультет.
<META name="author" content="Студент"> <META name="keywords" content="университет, студент, Карачаевск"> </HEAD> ... </HTML>	Автором документа является «Студент» Ключевые слова: университет, студент, Карачаевск

Элемент BODY

*Описание**:*** тело документа. В теле документа находится содержимое документа.

Начальный тег: ***не обязателен***

Конечный тег: ***не обязателен***

Определения атрибутов:

background = "url" (url – это строка, задающая путь в структуре каталогов до файла) – установка фоновой картинки

text = color (значение цвета может быть или 16-ричным числом (предваряемым знаком # в следующем формате #RRGGBB, где RR – градация красного цвета, GG – зеленого, BB – синего), или одним из следующих 16 названий цвета) - устанавливает цвет текста (для визуальных браузеров).

Black (Черный) = "#000000"	Green (Зеленый) = "#008000"
Silver (Серебро)= "#C0C0C0"	Lime (Известь) = "#00FF00"
Gray (Серый) = "#808080"	Olive (Оливковый) = "#808000"
White (Белый) = "#FFFFFF"	Yellow (Желтый)= "#FFFF00"
Maroon (Темно-бордовый) = "#800000"	Navy (Темно-синий) = "#000080"
Red (Красный) = "#FF0000"	Blue (Синий) = "#0000FF"
Purple (Фиолетовый) = "#800080"	Teal (Чирок) = "#008080"

Fuchsia (Фуксия) = "#FF00FF" Aqua (Аква) = "#00FFFF"

bgcolor = color – установка цвета фона документа

Пример:

<HTML> <HEAD> <TITLE> Колледж права и экономики </TITLE> </HEAD> <BODY bgcolor="white" text="#000000"> *... тело документа...* </BODY> </HTML>	Текст на странице будет отображаться черным цветом на белом фоне.

Задания для самостоятельного выполнения:

1. На жестком диске создать папку с именем HTML.
2. Открыть текстовый редактор "Блокнот".
3. В окне блокнота создать документ, печатающий в качестве заголовка документа название Вашей специальности.
4. Сохранить документ под именем index.html, обязательно с расширением **html** (или **htm**) в папке HTML.
5. Запустить любой из браузеров, установленный в системе, например, Internet Explorer (Пуск – Программы - Internet Explorer).
6. Используя меню Файл – Открыть, открыть в окне браузера свой файл и убедиться, что в строке заголовка напечатано название Вашей специальности.
7. Перейти в окно редактора Блокнот и добавить вывод в окне браузера «Физико-математический факультет».
8. Сохраните изменения.
9. Перейти в окно браузера. Сейчас, что бы просмотреть внесенные изменения, необходимо нажать кнопку "Обновить".
10. Используя метаданные, определить автора документа и ключевые слова: название Вашей специальности и название Вашего колледжа.
11. Установить цвет фона документа на свой вкус.
12. Установить фоновую картинку, для этого:

    12.1. на жестком диске найти файл с расширением jpg или gif;

12.2. скопировать найденный файл в свою папку;
12.3. установить фоновую картинку;
13. Изменить цвет текста документа на свой вкус.
14. Сохранить результаты, так как следующие задания опираются на результаты предыдущих.

Тема 2.2. Работа с отступами и шрифтами

Цели:

1. научиться использовать заголовки различных уровней;
2. научиться использовать абзацы, горизонтальные линии, «бегущие» строки;
3. научиться работать со шрифтами.

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

Заголовки: элементы H1, H2, H3, H4, H5, H6

Описание: описывают шесть уровней заголовков: от H1 (самый верхний) до H6 (самый нижний).
Начальный тег: ***необходим***
Конечный тег: ***необходим***
Определения атрибутов:
align = left|center|right|justify – выравнивание текста: left – выравнивание по левому краю, center – по центру (по умолчанию), right – по правому краю, justify – выравнивание по правому и левому краям.
Примеры:

<h1>Заголовок первого уровня</h1> <h2 align=”center”>Заголовок второго уровня</h2>	Печатается заголовок первого уровня. Печатается заголовок второго уровня, выровненный по центру.

Параграфы: элемент P

Описание: представляет параграф
Начальный тег: ***необходим***
Конечный тег: ***не обязателен***
Определения атрибутов:
align = left|center|right|justify – выравнивание текста
Примеры:

<p> Первый абзац. <p align=”left”>второй абзац.	Печатается первый абзац. Печатается второй абзац, выровненный по левому краю.

Форсирование обрыва строки: элемент BR

*Описание**:*** принудительно обрывает (оканчивает) текущую строку текста, но абзац не заканчивается.
Начальный тег: ***необходим***
Конечный тег: ***запрещён***
Пример:

<p> Первая строка абзаца Вторая строка абзаца.	Печатает абзац такого вида: Первая строка абзаца Вторая строка абзаца.

Изменение шрифта текста: элемент FONT

Описание: определяет вид, размер и цвет шрифта для текста.
Начальный тег: ***необходим***
Конечный тег: ***необходим***
Определения атрибутов:
size = number – устанавливает размер шрифта. Возможные значения:

- Целое число от 1 до 7. Устанавливает шрифт в определённый фиксированный размер, представление которого зависит от браузера пользователя.
- Относительное увеличение размера шрифта. "+1" означает: на один размер больше. "-3" означает: на три размера меньше. Все размеры находятся в пределах шкалы от 1 до 7

color = color – устанавливает цвет текста

face = string – задает имя шрифта
Пример:

 Строка	Печатает слово «Строка» черным цветом, шрифт «Tahoma» размером 1.

Элементы стиля шрифта: TT, I, B, BIG, SMALL, STRIKE, S и U

Описание:
TT: моноширинный текст, телетайп.
I: курсив.
B: полужирный.
BIG: "большой" шрифт.
SMALL: "малый" шрифт.
STRIKE и S: зачёркнутый текст.
U: подчёркнутый.
Начальный тег: ***необходим***
Конечный тег: ***необходим***
Примеры:

bold	**полужирный**
italic	*курсив*
bold italic	***полужирный курсив***
<tt>teletype text</tt>	телетайп
<big>big</big>	"большой" шрифт
small	"малый" шрифт

Подиндекс и надиндекс: элементы SUB и SUP

Описание: переводит текст в нижний и верхний регистр
Начальный тег: ***необходим***
Конечный тег: ***необходим***
Примеры:

H₂O	H_2O
E = mc²	$E=mc^2$

Бегущая строка: элемент MARQUEE

Описание: создания бегущей строки
Начальный тег: ***необходим***
Конечный тег: ***необходим***

Определения атрибутов:

bgcolor = color –цвет фона бегущей строки

loop = number – число повторов анимации бегущей строки (значение может быть целым или infinite – текст будет продолжать бегать пока читатель не перейдет на новую страницу)

direction = left|right|up|down – определяет направление: left (по умолчанию) – справа налево, right – слева направо, up – снизу вверх, down – сверху вниз.

scrollamount = number – скорость движения строки. (Рекомендуется ставить скорость "1", в этом случае строка выглядит более удобочитаемо и не дёргается).

scrolldelay = number – задаёт временной интервал между шагами бегущей строки.

width = number – ширина бегущей строки в пикселях.

height = number – высота бегущей строки. (Если вы делаете бегущую строку в одну строчку, то можно высоту не указывать, она сама подбирается под размер букв).

Примеры:

<marquee>Бегущая строка</marquee>	Надпись «Бегущая строка» двигается влево
<marquee direction=”right” scrollamount=1 scrolldelay=20>Бегущая строка</marquee>	Надпись «Бегущая строка» двигается вправо со скоростью равной 1 и с временным интервалом 20.

Горизонтальные линии: элемент HR

Описание: описывает горизонтальную линию

Начальный тег: ***необходим***

Конечный тег: ***запрещён***

Определения атрибутов:

align = left|center|right – определяет горизонтальное выравнивание линии.

size = number – определяет высоту линии.

width = number – определяет ширину линии. Значение может быть числовым и в процентах от ширины окна браузера.

color = color – устанавливает цвет линии.

Примеры:

<HR width="50%" align="center"> <HR size="5" width="100" align="left" color=”#ff0000”>	Линия, расположенная по центру и шириной 50% от размера окна. Красная линия, расположенная слева, высотой 5 точек и шириной 100 точек.

Задания для самостоятельного выполнения.

1. Запустить программу Блокнот и открыть документ index.html.
2. Изменить документ так, чтобы надпись «Физико-математический факультет» была заголовком первого уровня и выровнена по центру, надпись специальности – заголовком второго уровня, также выровнена по центру.
3. Добавить в документ параграф: Привет, меня зовут …. Сейчас мы изучаем язык HTML (Hypertext Markup Language).
4. Добавить в документ параграф: Мой адрес …, выровненный по правому краю.
5. Установить для слова Привет размер 5, цвет – желтый.
6. Для всего оставшегося параграфа: размер 4, цвет – maroon.
7. После слова привет вставить принудительный обрыв строки
8. Для последнего абзаца установить шрифт: размер 2, цвет – зеленый.
9. Для слова Привет установить шрифт Impact.
10. Для своей фамилии, имен, отчества установить шрифт Arial Black.
11. Подчеркнуть фамилию, имя, отчество.
12. Слово Привет написать курсивом.
13. Большим шрифтом выделить специальность.
14. Слово HTML зачеркнуть.
15. После слова HTML добавить предложение: Еще мы знаем немного математики, например, разность кубов вычисляется по формуле: $a^3-b^3=(a-b)*(a^2+ab+b^2)$, а сумма членов арифметической прогрессии равна: $S_n=(a_1+a_n)/2$.
16. Сделать строку Фамилия, имя, отчество бегущей, установив цвет фона gray, направление слева направо.
17. Нарисовать линию на всю ширину экрана под надписью «Моя первая страница»

Тема 2.3. Работа со списками

Цели:

1. *научиться работать со списками;*
2. *научиться структурировать документ.*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

Организация текста внутри документа. HTML позволяет определять внешний вид целых абзацев текста. Абзацы можно организовывать в списки, выводить их на экран в отформатированном виде или увеличивать левое поле. Разберем все по порядку.

Ненумерованные списки: <UL> ... </UL>.

Описание: описывают ненумерованный список.
Начальный тег: ***необходим***
Конечный тег: ***необходим***
Определения атрибутов:

type = *disc|circle|square* – задает стиль меток для данного списка: circle (кружок), disc (закрашенный кружок, по умолчанию) или square (квадрат).

Нумерованные списки: <OL> ... </OL>.

Описание: описывают нумерованный список.
Начальный тег: ***необходим***
Конечный тег: ***необходим***
Определения атрибутов:

type = *1|a|A|i|I* – задает стиль меток для данного списка: 1(арабские цифры, по умолчанию), a (латинский алфавит, нижний регистр) или A (латинский алфавит, верхний

регистр), i (римский алфавит, нижний регистр), I (римский алфавит, верхний регистр).

start = *number* – устанавливает номер первого элемента упорядоченного списка. По умолчанию это "1". Заметьте, что, хотя значением этого атрибута является целое число, соответствующие метки могут быть нечисловыми. Так, если стиль элемента списка это латинские буквы в верхнем регистре (A, B, C, ...), start=3 означает "C". Если стиль – это римские цифры в нижнем регистре, start=3 означает "iii" и т.д.

Каждый элемент обоих списков должен быть определен тэгом <LI>.

Элемент нумерованного и ненумерованного списков: <LI>.

Описание: описывают элемент списка.

Начальный тег: ***необходим***

Конечный тег: ***не обязателен***

Определения атрибутов:

value = *число* – устанавливает номер текущего элемента списка. Заметьте, что, хотя значением этого атрибута является целое число, соответствующие метки могут быть нечисловыми (см. атрибут **start**).

Примеры:

<UL type=”disc”> <LI> *первый элемент списка...*</LI> <LI> *второй элемент списка...* </UL>	• *первый элемент списка...* • *второй элемент списка...*
<OL type=”1”> <LI value=10> *первый элемент списка...* <LI> *второй элемент списка...* </OL>	10. *первый элемент списка...* 11. *второй элемент списка...*
<OL type=”i” start=”5”> <LI> *первый элемент списка...*</LI> <LI value=”10”> *второй элемент списка...* <LI> третий *элемент списка...* </OL>	*V.* *первый элемент списка...* *X.* *второй элемент списка...* третий *элемент списка...* *XI.* третий *элемент списка...*

Задания для самостоятельного выполнения.

1. Запустить программу Блокнот.
2. В новом документе оформить документ следующим образом (картинку можно вставить любую. Замечание: 1) найдите на винчестере нужную Вам картинку и скопируйте ее в свою папку, 2) файл назовите obomne.html и сохраните его в своей папке):

Кратко о себе:

1. Фамилия
2. Имя
3. Отчество
4. Дата рождения.
5. Место рождения.
6. Факультет.
7. Группа.
8. Хобби:

- Первый интерес
- Второй интерес
- Третий интерес

9. Знание компьютера:

- первая программа
- вторая программа
- третья программа

3. Для своей странички установить фон.
4. Строка «Кратко о себе» должна быть написана 7 шрифтом, цвет установить по своему усмотрению, шрифт – полужирный курсив и оформить ее в виде бегущей строки.
5. Строки, обозначенные цифрами, должны быть написаны 5 размером шрифта, цвет выбрать по своему усмотрению.
6. Хобби и название известных Вам компьютерных программ должны быть написаны 4 размером шрифта, цвет выбрать по своему усмотрению.
7. Под строкой Отчество провести линию синего цвета, ширина которой равна 3, во всю страницу.
8. Внизу страницы провести линию, под которой написать справа адрес своей электронной почты.

Тема 2.4. Работа со ссылками

Цели:

1. *научиться работать со ссылками;*
2. *научиться структурировать документ.*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

Основные сведения. Существует три типа ссылок: *внутристраничные* – они задают переходы в пределах одной страницы; *внутрисистемные* – ссылки между страницами в пределах одного и того же сервера; *межсистемные* – ссылки на страницы, расположенные на удаленных узлах web. Для определения ссылок предназначен специальный тег, который называется Anchor (якорь).

Элемент А.

Описание: определяет ссылку или якорь.
Начальный тег: ***необходим***
Конечный тег: ***необходим***
Определения атрибутов:

name = “строка” – именует текущий якорь, который может стать якорем назначения для другой гиперссылки. Значением этого атрибута должно быть уникальное имя якоря. Областью видимости этого имени является текущий документ.

href = “строка” – определяет размещение ресурса web, определяя таким образом ссылку с текущего элемента на якорь назначения, определённый этим атрибутом.

Примеры:

Информация	Создается метка с именем info.

ЧГПУ	Создается ссылка на сайт ЧГПУ.

Цвет ссылки можно задать при помощи атрибутов тэга **BODY**:

link = “color” – устанавливает цвет непосещённых гиперссылок.

vlink = “color” – устанавливает цвет посещённых гиперссылок.

alink = “color” – устанавливает цвет гиперссылок при выборе пользователем.

Пример:

<body link=”#ff0000” vlink=”#00ff00”>	Цвет непосещенных гиперссылок – красный, посещенных – синий.

Для указания ссылки на электронный ящик в значение атрибута href должно быть “mailto:имя_электронного_ящика”.

Пример:

Напишите мне	Создается ссылка «Напишите мне», щелкнув по которой, можно написать и отправить письмо.

Внутристраничные ссылки:

Создать имя (метку) для точки назначения, в которую должен осуществляться переход. Метка создается с помощью тега якоря, используя его атрибут NAME (например, <A NAME=”info”> Дополнительная информация </A>). Фраза "Дополнительная информация" при этом никак не будут выделены в тексте документа.

Для создания гиперссылки на эту точку документа используется тег <A> с атрибутом HREF=, при этом к имени якоря присоединяется знак #: <A HREF=”#info”> Просмотр дополнительной информации </A>. Такой фрагмент HTML-текста приведет к появлению в документе выделенного фрагмента (в нашем случае фразы Просмотр дополнительной информации), при нажатии на который произойдет переход к строчке Дополнительна информация.

1.1. Внутрисистемные ссылки:

В файле, в который мы хотим перейти, необходимо создать метку аналогично как при внутристраничных ссылках.

Создать гиперссылку, только имя якоря присоединяется к имени файла через знак # (Кратко о моих увлечениях и хобби можно посмотреть <A HREF=”obomne.html#info”> здесь </A>). При нажатии на выделенный фрагмент произойдет переход строчке #info в файле с именем obomne.html.

1.2. **Межсистемные ссылки.** Используя эти ссылки, можно установить связь с любой страницей на любом узле web.

1.2.1. В этом случае необходимо создать только гиперссылку, например, ссылка на сервер фирмы Microsoft выглядит следующем образом: *<A HREF="http://www.microsoft.com"> Переход на сервер компании Microsoft </A>*. При нажатии на выделенный фрагмент произойдет переход на сервер фирмы Microsoft.

Задания для самостоятельного выполнения:

1. В файл index.html добавить абзац «Кратко о моих увлечениях и хобби можно посмотреть здесь» после абзаца со словами «Еще мы изучаем математику …».
2. В начале файла obomne.html фразу «Кратко о себе» заключить в теги <A NAME=”info”> и <A>.
3. В документе main.html слово «здесь» оформить гиперссылкой на документ obomne.html на якорь «info»
4. Изменить цвет непосещенных гиперссылок на красный, а посещенных на черный.
5. Опробовать действие гиперссылки.
6. В файл obomne.html внизу страницы поместить абзац, состоящий из одного слова «Назад», выровненного по центру.
7. Организовать обратный переход.
8. В документ main.html добавить абзац «Здесь вы можете посмотреть сервер Карачаево-Черкесского государственного университета».
9. Оформить абзац по своему усмотрению.
10. Оформить абзац как гиперссылку на адрес http://www.кчгу.рф.
11. Адрес e-mail внизу страницы сделать ссылкой. Теперь пользователь сможет отправлять электронные письма по этому адресу.

12. В документе obomne.html добавить гиперссылку на сервер Челябинского государственного педагогического университета и отправку электронной почты.

Тема 2.5. Графика и мультимедиа

Цели:

1. *научиться работать с рисунками;*
2. *научиться структурировать документ с элементами мультимедиа.*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

1. Изображения *в HTML-документе.*

Вставка изображения в HTML-страницу осуществляется с помощью тэга <IMG>.

Тэг <IMG>.

Описание: внедряет изображение в текущий документ в месте определения элемента.
Начальный тег: ***необходим.***
Конечный тег: ***запрещен.***
Определения атрибутов:

src = “строка” – задающая путь в структуре каталогов до файла с изображением (наиболее распространённые форматы изображений: GIF, JPEG и PNG);

alt = “строка” – определяет альтернативный текст (который появляется при наведении курсора мыши на изображение);

width = “число” – определяет ширину объекта (изображения);

height = “число” – определяет высоту объекта. Для уменьшения времени загрузки страницы с графикой полезно указывать размер изображения. Если он известен еще до

загрузки страницы, то браузер может отвести рамку для картинки, а затем загружать текст на страницу. Пока загружается графика, посетитель страницы может начать читать текст.

border = “число” – определяет ширину рамки вокруг объекта;

align = “bottom|middle|top|left|right” – определяет позицию объекта по отношению к окружающему тексту (*bottom* – низ объекта должен быть выровнен вертикально по текущей базовой линии (по умолчанию), *middle* – центр объекта должен быть выровнен вертикально по текущей базовой линии, *top* - верх объекта должен быть выровнен вертикально по верхней границе текущей строки, *left* – прижимает объект к левому краю, *right* – прижимает объект к правому краю).

Примеры:

<img src=”pic1.jpg” width=”100” height=”100” align=”right”>	Вставить картинку pic1.jpg размером 100x100 и обтеканием текстом слева
<img src=”pic1.jpg” border=”2”>	Вставить картинку с рамкой шириной 2 пикселя.

Звук и видео.

Вы можете добавить на свою страничку звуки или видеоклипы. Они могут запускаться автоматически при загрузке странички. Для этого используется тег <EMBED>, который предназначен для встраивания объектов в документ.

Описание: внедряет объект текущий документ в месте определения элемента.

Начальный тег: ***необходим.***

Конечный тег: ***необходим.***

Определения атрибутов:

src = “строка” – задающая путь в структуре каталогов до файла с изображением (наиболее распространённые форматы изображений: GIF, JPEG и PNG);

autostart = “true|false” – определяет автозапуск звука или видеоклипа;

repeat = “true|false” – определяет после проигрывания записи начинать ли заново;

width = “число” – определяет ширину объекта (изображения);

height = “число” – определяет высоту объекта.

border = “число” – определяет ширину рамки вокруг объекта;
align = “center|left|right” – расположение пульта управления.

Примеры:

<EMBED SRC="media.wav" WIDTH=145 HEIGHT=55 autostart=true repeat=true></EMBED>	Проигрывать WAV файл со стандартным пультом управления, автостартом и с разрешением повтора

Задания для самостоятельного выполнения:

1. В графическом редакторе нарисовать эмблему Вашей специальности или факультета и сохранить его под именем emblem.jpg в Вашей папке.
2. В файл index.html вверху страницы по середине вставить картинку emblem.jpg и ее подписать «Эмблема факультета (или специальности) …», ширина рамки – 5 пикселов.
3. Внизу страницы расположить фотографию факультета (файл fmf.jpg). Установить соответствующую высоту, ширину и толщину рамки, выравнивание по левому краю и рядом по середине фотографию, сделать подпись «Физико-математический факультет».
4. В броузере отключить отображение графики (используя систему помощи броузера) и заново просмотреть созданный документ.
5. Нарисовать в графическом редакторе рисунок, в котором красиво написать «Назад».
6. Создать новый документ, назвав который photos.html, где разместите свою фотографию и сделать подпись к ней. Внизу страницы по середине сделать надпись «Назад», которую оформить в виде гиперссылки на документ index.html.
7. В документе index.html сделать гиперссылку на файл photos.html, добавив абзац «Здесь вы можете посмотреть мою фотографию».
8. В документе photos.html внизу страницы рядом с надписью «Назад» расположить картинку, которую оформить в виде гиперссылки на главную страницу.

Тема 2.6. Формы HTML и ввод данных

Цели:

Рассмотреть способы взаимодействия с пользователем и методы отправки пользовательских данных на web-сервер.

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

Формы HTML предназначены для организации взаимодействия с пользователем. Они позволяют вводить текст, осуществлять выбор из предложенных значений при помощи списков или кнопок. С помощью форм можно организовать интерактивный обмен информацией между web-страницей и сервером. Можно определить формы как электронные бланки для заполнения различных данных таких как, например, имя, возраст, выбор страны проживания и других. Как правило, форма работает совместно с установленным на сервере сценарным приложением, обрабатывающим введенную информацию.

Форма поиска

Одна из наиболее распространенных форм. Пользователь получает возможность ввести искомый запрос, определить область поиска и отправить запрос на сервер для обработки.

```
<html>
<body>
<form name="input" action="html_form_action.asp" method="get">
  <table border=1 bgcolor="#ddffdd">
    <tr>      <td align="center">
        <input type="text" name="search" size="50" value="Строка для поиска">
```

```
        <input type="submit" value="Поиск">
        <br>
        <input type="checkbox" name="news">Искать в новостях
        <input type="checkbox" name="arhive">Искать в архивах
       </td>    </tr>
  </table></form>
</body></html>
```

Отправка e-mail из формы

Вторая форма представляет собой форму заказа электронной рассылки, заполненная форма, отправляется по электронной почте для последующей обработки.

```
<html>
<body>
<form action="MAILTO:info@i.ru" method="post" enctype="text/plain">
  <h3>Подписка на новостную рассылку </h3>
  Имя:<br>
  <input type="text" name="name" value="ваше имя" size="20">
  <br>
  Mail:<br>
  <input type="text" name="mail" value="ваш e-mail" size="20">
  <br>
  Я хочу получать:
  <br>
  <input type="checkbox" name="news"> Новости о жизни
  <br>
  <input type="checkbox" name="courses"> Информацию о новых курсах
  <br>
  <input type="checkbox" name="books"> Информацию о новых изданиях
  <br>
  Комментарий:<br>
  <textarea rows="5" cols="30">
  </textarea>
  <hr>
  <input type="submit" value="Послать">
  <input type="reset" value="Сброс">
</form></body></html>
```

Формы

Форма является областью, которая может содержать элементы, позволяющие, пользователю вводить информацию (такие как текстовые поля, поля многострочного текста, раскрывающиеся меню, переключатели, флажки, и т.д.).

Форма определяется с помощью тегов ***<form></form>***, между которыми располагаются поля ввода, кнопки, а также все необходимые элементы оформления формы.

Тег <form> имеет ряд атрибутов, из которых необходимо выделить атрибуты action и method. Без этих атрибутов форма не сможет передать информацию от пользователя на сервер.

<form action="html_form_action.asp" method=get>

Атрибут ***Action*** указывает URL-адрес объекта, который должен получить данные формы.

Атрибут ***method*** может иметь два значения: ***get*** и ***post.***

Значение атрибута method=get заставляет web -браузер передать все данные формуляра по URL-адресу, заданному в action. При этом введенные при заполнении формы данные просто добавляются в адресную строку с использованием разделителя – знака вопроса. Этот метод удобен для небольших форм.

Значение атрибута ***method=post*** заставляет web -браузер, прежде всего, связаться с сервером, обрабатывающим форму, и только после установки связи приступить к передаче данных, для обработки которых будут использоваться специальные сценарии.

Поля ввода

Большинство элементов ввода и управления в форме можно описать при помощи тега ***<input>***, обязательными для которого являются атрибуты name (приписывает данному элементу ввода уникальное имя, использующееся для дальнейшей обработки формы) и type (определяет тип элемента управления или ввода).

Текстовые поля

Текстовое поле ***(type=text)*** определяет однострочное поле ввода и используется, когда необходимо, чтобы пользователь ввел в форму данные в произвольной форме но ограниченные по объему (слова, словосочетания, числа и т.д.).Следующий пример демонстрирует простейшую форму для ввода имени и фамилии.

```
<html><body><form>
  Имя:
  <input type="text" name="firstname">
```

```
 <br>
 Фамилия:
 <input type="text" name="lastname">
</form></body></html>
```

В большинстве браузеров по умолчанию ширина текстового поля равна 20 символам. Для изменения этого значения используется атрибут size. В следующем примере поле для ввода имени ограничено 10 символами.

```
<html><body><form>
 Имя:
 <input type="text" name="firstname" size="10">
 <br>
 Фамилия:
 <input type="text" name="lastname">
</form></body></html>
```

Здесь необходимо отметить, что атрибут size ограничивает только видимую область ввода данных, а не длину вводимой строки. Для этой цели используется атрибут maxlength. В следующем примере максимальное количество символов для ввода имени ограничено тремя.

```
<html><body><form>
 Имя:
 <input type="text" name="firstname" maxlength="3">
 <br>
 Фамилия:
 <input type="text" name="lastname">
</form></body></html>
```

При вводе в поле имени слова "Елена" форма воспримет только первые три символа имени:

Кроме того, при использовании текстовых полей возможно задать значения по умолчанию. Для этого используется атрибут value. Применение этого атрибута показано на следующем примере:

```
<html>
<body>
<form>
 Имя:
 <input type="text" name="firstname" value="Билл">
 <br>
 Фамилия:
```

```
  <input type="text" name="lastname" value="Гейтс">
</form>
</body>
</html>
```

Необходимо отметить, что применение атрибутов для различных типов полей вывода и управления может отличаться. Так, например, атрибут size для текстовых полей ввода (type="text" или type="password") указывает максимальное количество символов, отображаемых в поле, а для прочих элементов – занимаемый по горизонтали размер в пикселях.

Поле пароля

Поле пароля (***type=password***) создает защищенное поле ввода, которое дает возможность пользователю, заполняющему форму ввести текст, но в отличие от обычного текстового поля, вводимые данные при отображении на мониторе заменяются звездочками или точками.

Следующий пример демонстрирует простейшую форму для ввода имени пользователя и пароля.

```
<html><body><form>
  Имя пользователя:
  <input type="text" name="user" value="Елена">
  <br>
  Пароль:
  <input type="password" name="password" value="tktyf">
</form></body></html>
```

Необходимо обратить внимание, что хотя значение пароля и задано по умолчанию, при отображении браузер выводит вместо символов звездочки.

Переключатели

Переключатели или радиокнопки (***type=radio***) определяют поля выбора одного значения из нескольких доступных. Поля этого типа часто используются в диалоговых окнах. Для каждой позиции переключателя создается свой тег <input type=radio>. Группируются переключатели при помощи одинакового имени, задаваемого атрибутом name.

```
<html><body><form>
  Укажите Ваш пол:  <br>
  <input type="radio" name="sex" value="male"> мужчина
  <br>
  <input type="radio" name="sex" value="female"> женщина
```

```
</form></body></html>
```

Необходимо отметить, что в отличие от текстового поля и поля пароля атрибут value задает значение, которое будет передано серверу для дальнейшей обработки в случае выбора данного переключателя. Для выбора по умолчанию одного из возможных значений группы переключателей используется атрибут checked.

```
<html>
<body
<form>
  Укажите Ваш пол:
  <br>
  <input type="radio" name="sex" value="male" checked="checked">
мужчина
  <br>
  <input type="radio" name="sex" value="female"> женщина
</form>
</body>
</html>
```

В данном примере по умолчанию выбран мужской пол:

Флажки

Флажки ***(type=checkbox)*** используются, когда необходимо, чтобы пользователь выбрал один или несколько вариантов из ограниченного числа вариантов выбора. Флажки в форме не зависят друг от друга, их можно установить или сбросить в любой комбинации. Для каждого флажка необходимо задать свое уникальное имя при помощи атрибута name. Создание двух флажков с одним именем не вызовет ошибки при отображении формы, но не позволит сценарию обработки на сервере корректно обработать передаваемые с формы данные.

```
<html>
<body>
<form>
  В этом году я собираюсь приобрести:
  <br>
  <input type="checkbox" name="computer">
  Компьютер
  <br>
  <input type="checkbox" name="notebook">
  Ноутбук
  <br>
```

```
  <input type="checkbox" name="printer">
  Принтер
  <br>
  <input type="checkbox" name="scanner">
  Сканер
</form>
</body>
</html>
```

При помощи атрибута checked можно установить, какие из флажков будут выбраны по умолчанию при загрузке страницы. Отличие от переключателей заключается только в том, что для флажков можно отметить сразу несколько вариантов.

```
<html>
<body>
<form>
  В этом году я собираюсь приобрести:
  <br>
  <input type="checkbox" name="computer" checked="checked">
  Компьютер
  <br>
  <input type="checkbox" name="notebook">
  Ноутбук
  <br>
  <input type="checkbox" name="printer">
  Принтер
  <br>
  <input type="checkbox" name="scanner" checked="checked">
  Сканер
</form>

</body>
</html>
```

В данном примере по умолчанию выбраны флажки "Компьютер" и "Сканер":

При отправке данных формы с флажками на сервер выбранным флажкам присваивается значение по умолчанию "on". Как правило, этого достаточно для корректной обработки данных, но в некоторых случаях удобнее задать каждому флажку свое значение при помощи атрибута value.

<input type="checkbox" name="printer" value="Принтер">

Командные кнопки

Командная кнопка отправки (***type=submit***) используются для выполнения пересылки данных формы на сервер. Метод отправки и адрес файла, обрабатывающего полученную информацию задаются в теге <form> при помощи атрибутов method и action. Командная кнопка сброса (***type=reset***) возвращает форму к исходному состоянию (очищает форму). При этом данные не передаются.

В следующем примере показана форма поиска с двумя кнопками отправки и сброса.

```
<html>
<body>
 <form name="input" action="html_form_action.asp" method="get">
  Найти:
  <input type="text" name="search" size=25>
  <br>
  <input type="submit">
  <input type="reset">
</form>
</body></html>
```

Если ввести в текстовое поле какие-то символы и нажать кнопку "Подача запроса", то введенная информация будет послана на страницу с именем "html_form_action.asp". При нажатии на кнопку "Сброс" текстовое поле очистится.

Надписи на кнопках "Подача запроса" и "Сброс" установлены по умолчанию. Для их изменения необходимо использовать атрибут value.

Кроме кнопок отправки и сброса существует также возможность добавлять пользовательские кнопки (type=button), которые могут использоваться для выполнения процедур (скриптов) непосредственно на Web-странице.

Поле выбора файла

Поле выбора файла (***type=file***) создает поле для выбора файла, который будет загружен на сервер вместе с информацией формы. Рядом с полем ввода отображается командная кнопка "Обзор…", открывающая стандартное диалоговое окно выбора файла.

```
<html>
<body>
```

```
<form>
  Прикрепить файл:
  <br>
  <input type="file" size="50">
</form>

</body>
</html>
```

Для поля выбора файла по аналогии с текстовым полем можно использовать атрибуты size, maxlength, value.

Списки выбора

Списки выбора бывают двух типов: раскрывающиеся списки (выпадающие меню) и списки с множественным выбором. Независимо от типов списков описываются они одинаково с помощью пары тегов ***<select> </select>***. Отдельные элементы списка задаются с использованием тега <option>. Тип списка определяется при помощи атрибутов тега ***<select>.***

Атрибут name задает имя поля для отправки выбранных пунктов списка на сервер. Атрибут multiple разрешает множественный выбор. Атрибут size определяет, какое количество пунктов списка будет одновременно отображено на экране. При этом, если атрибут multiple не задан и size=1, то на экране отображается раскрывающийся список, если же задан атрибут Multiple или значение size больше 1, то список отображается развернутым.

Раскрывающийся список выбора

Раскрывающиеся списки выбора по своему назначению соответствуют переключателям, в то же время их использование предпочтительно в тех случаях, когда количество вариантов выбора достаточно велико. Как правило, при выборе более чем из трех вариантов желательно вместо переключателей использовать раскрывающиеся списки.

В следующем примере создан простой раскрывающийся список выбора ноутбука по производителю.

```
<html>
<body>
<form>
  Выбор ноутбука по производителю:
  <select name="notebook">
```

```
    <option value="aser">Aser
    <option value="asus">Asus
    <option value="compaq">Compaq
    <option value="hp">HP
    <option value="sony">Sony
    <option value="toshiba">Toshiba
  </select>
</form>
</body>
</html>
```

Необходимо заметить, что по умолчанию выбирается первое значение из списка. При помощи атрибута ***selected*** тега <option> это значение можно изменить. Следующий пример показывает раскрывающийся список выбора размера экрана ноутбука с предварительно установленным значением "15.4".

```
<html>
<body>
<form>
  Выбор размера экрана ноутбука
  <select name="tft">
    <option value="tft-12">12"
    <option value="tft-13">13"
    <option value="tft-14">14"
    <option value="tft-15">15"
    <option value="tft-15-4" selected="selected">15.4"
    <option value="tft-17">17"
  </select>
</form>
</body>
</html>
```

Развернутый список выбора

Развернутые списки используются, как правило, при необходимости множественного выбора значений. В следующем примере раскрывающийся список выбора ноутбука по производителю преобразован в развернутый список с возможностью множественного выбора.

```
<html>
<body>
<form>
```

```
  Выбор ноутбука по производителю:
  <select name="notebook" multiple>
    <option value="aser">Aser
    <option value="asus">Asus
    <option value="compaq">Compaq
    <option value="hp">HP
    <option value="sony">Sony
    <option value="toshiba">Toshiba
  </select>
</form>
</body>
</html>
```

Необходимо отметить, что для списков с множественным выбором значение атрибута ***size*** по умолчанию устанавливается равным 4, кроме того, ни одно из значений не является выбранным по умолчанию. В следующем примере показана форма с развернутым списком высотой в 6 пунктов с предварительно заданными значениями "Compaq" и "Sony".

```
<html>
<body>
<form>
  Выбор ноутбука по производителю:
  <select name="notebook" multiple size=6>
    <option value="aser">Aser
    <option value="asus">Asus
    <option value="compaq" selected="selected">Compaq
    <option value="hp">HP
    <option value="sony" selected="selected">Sony
    <option value="toshiba">Toshiba
  </select></form></body>
</html>
```

Текстовая область

В отличие от текстового поля ***<input type=text>*** текстовая область позволяет вводить многострочный текст большого объема. Такие области часто используются при вводе сообщений, комментариев.

Текстовое поле описывается при помощи тегов ***<textarea> </textarea>***, между которыми можно разместить предварительно

отформатированный стандартный текст. Атрибуты ***cols*** и ***rows*** задают размер видимой области текстового поля.

В следующем примере создано текстовое поле с предварительно введенным текстом.

```
<html>
<body>
  <textarea rows="7" cols="30">
```

В данном примере мы создали текстовую область с шириной в 30 символов и высотой в 7 строк.

Заданное значение высоты не ограничивает общий объем вводимого текста, а влияет только на размер отображаемой на экране текстовой области.

Для просмотра всего текста необходимо воспользоваться полосой прокрутки.

```
  </textarea>
</body></html>
```

Внешний вид форм

Существует различные способы оформления внешнего вида форм. Один из них основан на использовании пары тегов ***<fieldset> </fieldset>,*** которые позволяют объединить тематически близкие и рядом расположенные поля формы в группу и выделить ее визуально. Другой способ основан на применении стандартных средств форматирования HTML, включая цветовое и шрифтовое оформление, графику, таблицы и т.п.

Группировка полей формы

Теги <fieldset> </fieldset> объединяют поля формы в группы, выделяют их визуально, упрощают навигацию в форме с помощью клавиши [Tab] (в первую очередь обходятся поля в пределах группы). Для того чтобы объединить несколько элементов ввода или управления в группу достаточно поместить их внутри тегов <fieldset> </fieldset>. Закрывающий тег </fieldset> обязателен.

В следующем примере создается форма, содержащая поля для ввода имени и фамилии, объединенные в группу, а также кнопку отправки

```
<html>
<body>
<form name="input" action="html_form_action.asp" method="get">
  <fieldset>
    Имя:
```

```
    <input type="text" name="firstname" value="Билл">
    <br>
    Фамилия:
    <input type="text" name="lastname" value="Гейтс">
  </fieldset>
  <input type="submit" value="отправить">
</form>
</body>
</html>
```

При помощи пары тегов ***<legend> </legend>*** образованной группе полей можно присвоить имя. В следующем примере форма ввода дополнена полями для ввода информации о месте работы и занимаемой должности, объединенными во вторую группу. Каждой группе полей присвоен свой заголовок.

```
<html>
<body>
<form name="input" action="html_form_action.asp" method="get">
  <fieldset>
    <legend>Личные данные</legend>
    Имя:
    <input type="text" name="firstname" value="Билл">
    <br>
    Фамилия:
    <input type="text" name="lastname" value="Гейтс">
    <br>
    Пол:
    <input type="radio" name="Sex" value="Male" checked="checked"> Мужской
    <input type="radio" name="Sex" value="Female"> Женский
  </fieldset>
  <fieldset>
    <legend>Данные о работе</legend>
    Место работы:
    <input type="text" name="work" value="Microsoft">
    <br>
    Должность:
    <input type="text" name="status" value="Президент">
  </fieldset>
  <input type="submit" value="отправить">
```

```
</form>
</body>
</html>
```

Оформление форм стандартными средствами HTML

Несмотря на простоту использования тегов группировки <fieldset>, их возможности по оформлению форм весьма ограничены. Чаще всего для оформления внешнего вида форм используются стандартные средства HTML. В следующем примере показано, как с помощью таблицы поля формы выровнены друг под другом, а заголовок формы и кнопка отправки размещены по центру формы. Сама форма визуально выделена желтым цветом.

```
<html>
<body>
<form name="input" action="html_form_action.asp" method="get">
  <table bgcolor="#FFFF99">
    <tr>
      <th colspan="2" align="center">Личные данные:</td>
    </tr>
    <tr>
      <td align="right">Имя:</td>
      <td align="left"><input type="text" name="firstname" value="Билл"></td>
    </tr>
    <tr>
      <td align="right">Фамилия:</td>
      <td align="left"><input type="text" name="lastname" value="Гейтс"></td>
    </tr>
    <tr>
      <td colspan="2" align="center"><input type="submit" value="отправить"></td>
    </tr>
  </table>
</form></body></html>
```

Раздел III. Каскадные таблицы стилей

Тема 3.1. Предназначение стилей. Синтаксис CSS и способы их подключения

Цели:

1. *Рассмотреть предназначение и область применения CSS.*
2. *Рассматреть грамматику CSS и способы подключения таблиц стилей.*
3. *Привести примеры использования CSS. .*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

Предназначение стилей

Консорциум W3C (World Wide Web Consortium) — некоммерческая организация, ответственная за стандартизацию HTML — создала при разработке стандарта HTML 4.0 каскадные таблицы стилей.

Таблицы стилей определяют вывод элементов HTML (как, например, тег font и атрибут color в HTML 3.2). Каскадные таблицы стилей обычно хранятся во внешних файлах с расширением .css. Этот способ удобен тем, что позволяет изменить внешний вид и компоновку всех необходимых страниц в web, редактируя только один, единственный документ CSS.

Технология CSS позволяет разработчикам одновременно управлять стилем и компоновкой множества web-страниц, определять стиль для каждого элемента HTML и применять его к любому

количеству web-страниц. Также CSS дает возможность из одного документа ссылаться на несколько внешних таблиц стилей. Таким образом, чтобы сделать глобальное изменение, надо просто изменить стиль, и все элементы в web будут автоматически изменены.

Разработчик может определять стили следующими способами:

внутри единственного элемента HTML;

внутри элемента <head> страницы HTML;

во внешнем файле CSS.

Приоритет использования стилей

Для каскадных таблиц стилей определен приоритет использования. Если для элемента HTML определено более одного стиля, то все стили будут последовательно "каскадированы" в новую "виртуальную" таблицу стилей, согласно следующим правилам:

стили, используемые по умолчанию браузером;

стили, хранящиеся во внешней таблице;

стили, хранящиеся во внутренней таблице стилей (внутри тега <head>);

встроенный стиль (внутри элемента HTML).

Эти правила определяют порядок увеличения приоритета стилей.

Таким образом, встроенный стиль (внутри элемента HTML) имеет наивысший приоритет, т.е. будет переопределять стиль, который объявлен в теге <head>, во внешней таблице стилей или в браузере (значение по умолчанию).

Синтаксис

Синтаксис CSS состоит из трех частей: селектора, свойства и значения:

селектор {свойство: значение}

Селектор — элемент/тег HTML, который необходимо определить. Свойство — атрибут, который желательно изменить. Каждое свойство может принимать значение. Существует ряд правил и рекомендаций, которые необходимо запомнить.

- Свойство и значение разделяются двоеточием и помещаются внутри фигурных скобок:

p {font-size: 75%}

- Если значение состоит из нескольких слов, то необходимо поместить значение в кавычки:

h1 {font-family: "lucida calligraphy"}

- Если требуется определить более одного свойства, то необходимо разделить свойства точкой с запятой:

table { font-family: arial, "sans serif"; border-style: dotted}

- Чтобы определения стилей было удобно читать, можно каждое свойство писать на отдельной строке:

h2 {font-family: arial; margin-right: 20pt; color:#ffffff }

- При определении правил допускается группировка селекторов, при этом в качестве разделителя селекторов используется запятая. В следующем примере в группу были объединены все элементы заголовков. Все элементы заголовков будут выведены шрифтом sans serif:

p,table,li

{font-family: "sans serif";}

Селектор класса (class)

При web-разработке часто возникает необходимость в задании различных стилей для одного и того же типа элемента HTML. Для решения этой задачи используют селектор класса.

Предположим, что в документе требуется два типа заголовков: более крупный заголовок должен иметь внешний отступ 10 пунктов, а второй — 20 пунктов. Вот как это можно сделать с помощью стилей:

h1.stepleft {margin-left: 10pt}

h2.stepright {margin-left: 20pt}

Для применения этих стилей в документе HTML необходимо будет использовать атрибут класс:

<h1 class="stepleft">

Заголовок с внешним отступом 10 пунктов.

</h1>

<h2 class="stepright">

Заголовок с внешним отступом 20 пунктов.

</h2>

В то же время не допускается следующее определение атрибута class:

< h1 class="stepleft" h2 class="stepright">,

т.е. можно определить только один атрибут class

Можно также опустить имя тега в селекторе, чтобы определить стиль, который будет использоваться всеми элементами HTML, имеющими определенный class.

.left {margin-left: 40pt}

В следующем примере все элементы HTML, имеющие class="left", будут иметь внешний отступ, равный 40 пунктам.

В следующем коде элементы table и p имеют class="left". Это означает, что оба элемента будут следовать правилам в селекторе ".left":

<table class="left">

Эта таблица будет иметь внешний отступ, равный 40 пунктам.

</table>

Этот параграф будет иметь внешний отступ, равный 40 пунктам.

Для корректности обработки кода браузерами не рекомендуется начинать имя класса с числа, т.к. это не будет работать в Mozilla/Firefox.

Селектор идентификатора (id)

Стили элементов HTML можно определить также с помощью селектора идентификатора, который определяется символом #.

Следующее правило стиля будет применимо к элементу, который имеет атрибут id со значением "fontsz":

```
#fontsz{font-size: 50%}
```

Следующее правило стиля будет применимо к элементу ul, который имеет атрибут id со значением "first":

```
ul#first
{
list-style: disc;
color: #ffffff
}
```

Для корректности обработки кода браузерами не рекомендуется начинать имя ID с числа, т.к. это не будет работать в Mozilla/Firefox.

Комментарии CSS

Для пояснения кода и для последующего его редактирования используются комментарии, которые игнорируются браузерами. Комментарий CSS начинается символом "/*" и заканчивается символом "*/", как в примере ниже:

```
 /* Первый комментарий. */
table
{
/* Второй комментарий. */
margin-left: 10pt;;
font-family: "sans serif";
/* Третий комментарий. */
```

```
border-style:outset
}
```

Подключение таблицы стилей

Когда браузер считывает таблицу стилей, он форматирует документ согласно с этой таблицей. Существует три способа подключения таблицы стилей.

1. **Подключение внешней таблицы стилей**

Этот способ обычно применяется в ситуациях, когда один стиль определяется для множества страниц. Если необходимо быстро поменять дизайн web-сайта этот способ идельно подходит – редактируется только один файл с таблицей стилей. Для подключения внешней таблицы стилей предназначен тег <link>, который размещается в разделе заголовка:

```
<head>
<link rel="stylesheet" type="text/css"
href="first.css" />
</head>
```

Данный код указывает браузеру на то, что определение стиля хранятся в файле first.css. Последний, в свою очередь, выполнит форматирование документа согласно с этим определением стиля.

Некоторые замечания:

1. внешнюю таблицу стилей можно создать в любом текстовом редакторе;
2. файл с внешней таблицей стилей не должен содержать никаких тегов html;
3. файл с внешней таблицей стилей необходимо сохранить с расширением .css.

Пример:

```
ul {margin-left: 10pt}
li {font-family: arial}
body {background-color: blue}
```

Для корректности отображения браузерами таблиц стилей необходимо исключить пробелы между значением свойства и единицами измерения, т.к. это будет работать правильно только в IE6, но не в Mozilla/Firefox или Netscape.

Пример с пробелом — "margin-left: 20 px";

пример без пробела — "margin-left: 20px".

2. **Подключение внутренней таблицы стилей**

Данный способ применяется тогда, когда один документ имеет уникальный стиль. Для определения внутренних стилей используют тег <style>, который размещается в разделе заголовка:

```
<head>
<style type="text/css">
li {font-family: arial}
p {font-size: 25%}
h1 {margin-right: 10pt}
</style>
</head>
```

В соответствии с определениями стиля браузер форматирует документ. Однако если он встретит незнакомые теги, то проигнорирует их. Например, старый браузер, который не поддерживает стили, проигнорирует тег <style>, но содержимое последнего будет выведено на странице. Чтобы этого не произошло, необходимо скрыть описание стиля в элементе комментария HTML:

```
<head>
<style type="text/css">
<!--
body {background-color: green}
table {border-style: dotted}
-->
</style>
</head>
```

3. Встроенные стили

Данный способ используется в том случае, если необходимо применить стиль к одному появлению элемента. Метод обладает существенным недостатком, т.к. он смешивает содержимое документа с его представлением и теряет, соответственно, многие преимущества таблиц стилей.

Для определения встроенных стилей в соответствующем теге применяют атрибут style, который может содержать любое свойство CSS.

Пример:

```
<body style="background-color: blue" >
<h1 style="color: yellow" >Заголовок H1</h1>
</body>
```

В примере показано, как задать цвет фона документа и цвет заголовка h1.

Несколько таблиц стилей

В случае объединения или использования нескольких таблиц стилей могут возникнуть ситуации, когда некоторые свойства заданы для одного и того же селектора в разных таблицах стилей; тогда будут использованы значения из более конкретной таблицы стилей.

Например, внешняя таблица стилей имеет следующие свойства для селектора p:

```
p {font-family: "sans serif";
margin-left: 10pt;
font-style: italic;
font-weight: bold}
```

А внутренняя таблица стилей имеет для селектора p следующие свойства:

```
p{margin-left: 20pt;
font-style: normal;
font-weight: bolder}
```

Итак, при соединении внешней таблицы стилей с внутренней для селектора p будут определены следующие свойства:

```
font-family: "sans serif";
margin-left: 20pt;
font-style: normal;
font-weight: bolder
```

Таким образом, тип шрифта будет наследоваться из внешней таблицы стилей, а его стиль, вес и внешний отступ параграфа определяются внутренней таблицей стилей.

Тема 3.2. Параметры CSS для фона. Параметры текста в CSS.

Цели:

1. *Рассмотреть синтаксис, методы и свойства задания фона.*
2. *Рассмотреть механизмы представления текстовой информации: оформление, выравнивание, отступы и др.*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.

2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

Параметры CSS для фона

Параметры CSS для фона определяют фоновые эффекты элемента. Параметры фона в CSS позволяют управлять цветом фона элемента, задавать в качестве фона изображение, повторять циклически фоновое изображение вертикально или горизонтально и позиционировать изображение на странице.

Параметр 'background'

Этот параметр предназначен для задания всех свойств фона в одном объявлении.

Наследование: нет.

Примеры:

body{ background: url(picture.gif)}

body{ background: url(speciality/image.gif) repeat scroll}

body{background: yellow url(image.gif) repeat-x bottom}

Может принимать следующие значения: background-color, background-repeat, background-image, background-attachment background-position.

Параметр 'background-attachment'

Этот параметр определяет, будет ли фоновое изображение зафиксировано в определенном месте или будет перемещаться вместе со всей страницей.

Наследование: нет.

Может принимать следующие значения: Scroll, Fixed.

Scroll: Фоновое изображение перемещается, когда перемещается страница.

Fixed: Фоновое изображение не перемещается, когда перемещается страница.

Пример:

body { background-attachment: fixed;background-repeat: repeat }

Параметр 'background-color'

Этот параметр задает фоновый цвет элемента.

Наследование: нет.

Может принимать следующие значения: color, transparent.

Color: Значение color может быть названием цвета (red), значением rgb (rgb(255,0,0)) или шестнадцатеричным числом (#ff0000)

Transparent: Фоновый цвет является прозрачным

Пример:

h1{ background-color: gray;font-family: arial}

Параметр 'background-image'

Данный параметр задает изображение в качестве фона.

Наследование: нет.

Может принимать следующие значения: url, none.

url: Путь доступа к изображению

none: Фонового изображения нет

Пример:

body

{ background-image: url(image.gif);

background-repeat: repeat;

background-attachment: fixed}

В тех случаях, когда изображение недоступно, необходимо задать цвет background-color, который будет использован.

Параметр 'background-position'

Этот параметр задает начальное положение фонового изображения.

Наследование: нет.

Может принимать следующие значения:

Top left, top center, top right, enter center, center left, right bottom, left bottom, bottom center, bottom right: Если определить только одно ключевое слово, то вторым значением подразумевается "center.

x-% y-%: Первое значение является горизонтальной координатой, второе значение — вертикальной. Верхний левый угол — 0% 0%. Правый нижний угол — 100% 100%. Если определено только одно значение, то вторым значением подразумевается 50%.

x-pos y-pos: Первое значение является горизонтальной координатой, второе значение — вертикальной. Верхний левый угол — 0 0. Единицами измерения могут быть пиксели (0px 0px) или любые другие единицы измерения CSS. Если определено только одно значение, то вторым значением подразумевается 50%. Можно смешивать % и эти координаты.

Пример:

```
Body {
background-image: url(http://www.intuit.ru/speciality/image.gif);
background-position: right top;
background-attachment: fixed}
body{
background-image: url(http://www.intuit.ru/speciality/image.gif);
background-repeat: no-repeat;
background-position: 100% 100%;
background-attachment: fixed}
```

Параметр 'background-repeat'

Этот параметр определяет, каким образом будет повторяться фоновое изображение.

Наследование: нет.

Может принимать следующие значения:

Repeat: Фоновое изображение будет повторяться по вертикали и по горизонтали.

Repeat-x: Фоновое изображение будет повторяться по горизонтали

Repeat-y: Фоновое изображение будет повторяться по вертикали

No-repeat: Фоновое изображение будет выведено только один раз

Пример:

```
body{
background-image: url( image.gif);
```

background-repeat: repeat;

background-attachment: fixed}.

Параметры текста в CSS.

Параметры текста CSS позволяют управлять внешним видом текста. Можно изменять цвет текста, увеличивать или уменьшать интервал между символами, выравнивать текст, оформлять текст, делать отступ для первой строки текста и т.д.

Параметр 'color'

Этот параметр задает цвет текста.

Наследование: да.

Может принимать следующие значения:

Color: Значением color может быть название цвета (red), значение rgb (rgb(255,0,0)) или шестнадцатеричное значение (#ff0000).

Пример:

h1{ color: green}

Параметр 'direction'

Параметр задает направление текста.

Наследование: да.

Может принимать следующие значения:

Ltr: Направление текста слева направо.

Rtl: Направление текста справа налево.

Пример:

p{ direction: rtl}

Параметр 'letter-spacing'

Данный параметр увеличивает или уменьшает интервал между символами.

Наследование: да.

Примечание: допускаются отрицательные значения.

Может принимать следующие значения:

Normal: Определяет обычный пробел между символами.

Length: Определяет фиксированный пробел между символами.

Примеры:

Pre { letter-spacing: -2px }

pre{ letter-spacing: 20px}

Параметр 'text-align'

Этот параметр задает выравнивание текста в элементе.

Наследование: да.

Может принимать следующие значения:

Left: Выравнивает текст слева;

Right: Выравнивает текст справа;

Center: Центрирует текст;

Justify: Выравнивает текст по ширине.

Пример:

h1{ text-align: right}

Параметр 'text-decoration'

Данный параметр задает дополнительное оформление текста.

Наследование: нет.

Примечание: цвет оформления должен быть задан свойством "color".

Может принимать следующие значения:

None: Определяет обычный текст

Underline: Определяет линию под текстом

Overline: Определяет линию над текстом

Line-through: Определяет линию через текст

Blink: Определяет мигающий текст

Пример:

h1{ text-decoration: overline}

Параметр 'text-indent'

Данный параметр создает отступ для первой строки текста элемента.

Наследование: да.

Примечание: допускаются отрицательные значения, для которых первая строка будет сдвинута влево.

Может принимать следующие значения:

Length: Определяет фиксированный отступ;

%: Определяет отступ в % от ширины родительского элемента;

Примеры:

pre{ text-indent: -10px}

p{ text-indent: 10px}

Параметр 'text-transform'

Этот параметр управляет регистром символов в элементе.

Наследование: да.

Может принимать следующие значения:

None: Определяет обычный текст с символами нижнего регистра и заглавными буквами;

Capitalize: Каждое слово в тексте начинается с заглавной буквы;

Uppercase: Определяет только заглавные буквы:

Lowercase: Определяет только символы нижнего регистра.

Примеры:

h1{text-transform: capitalize}

pre{ text-transform: lowercase}

Параметр 'white-space'

Параметр задает способ обработки пробелов внутри элемента.

Наследование: да.

Может принимать следующие значения:

Normal: Браузер игнорирует пробел;

Pre: Браузер сохраняет пробел. Действует как тег <pre> в HTML;

Nowrap: Текст не будет переноситься на другую строку, пока не встретится тег <br>.

Пример:

pre{white-space: pre}

Параметр 'word-spacing'

Данный параметр увеличивает или уменьшает пробел между словами.

Наследование: да.

Примечание: допускаются отрицательные значения.

Может принимать следующие значения:

Normal: Определяет обычный пробел между словами;

Length: Определяет фиксированный пробел между словами.

Примеры:

Pre { word-spacing: -10px}

h1{ word-spacing: 35px}.

Тема 3.3. Шрифты в CSS

Цели:

1. *Рассмотреть общие способы и методы применения шрифтов, их характеристики и атрибуты.*
2. *Привести примеры работы со шрифтами в CSS.*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

Параметры шрифта в CSS позволяют изменить семейство шрифта, степень жирности, размер, и стиль текста. Обратите внимание на то, что в CSS1 шрифты идентифицируются названием шрифта. В случае, если браузер не поддерживает данный шрифт, будет использоваться шрифт по умолчанию.

Параметр 'font'

Параметр font предназначен для сокращенной записи, позволяющей задать все свойства шрифта в одном объявлении (за исключением некоторых системных шрифтов).

Наследование: да.

Может принимать следующие значения:

Font-style, font-variant, font-weight, font-size* /line-height, font-family: Задает свойства шрифта. Значение line-height задает расстояние между строками. Это значение может быть числом, % или размером;

Caption: Определяет шрифт для элементов управления с заголовком (таких, как кнопки, раскрывающиеся меню и т.д.);

Icon: Определяет шрифты, которые используются метками пиктограмм приложений;

Menu: Определяет шрифты, которые используются для раскрывающихся меню;

Message-box: Определяет шрифты, которые используются в диалоговых окнах;

Small-caption, status-bar: Определяет шрифты, которые используются в строке состояния окна.

Примеры:

pre{font: oblique arial}

h1{font: italic 200 arial}

p{font: italic small-caps 10px/12px serif}

p{font: status-bar}

Параметр 'font-family'

Данное свойство определяет для элемента список имен семейств шрифтов и/или имен порождающих семейств в порядке их приоритета. Если специфицировано несколько значений, то браузер будет использовать первое распознанное значение.

Существует два типа значений font-family:

family-name: имя семейства шрифтов ("times", "courier", "arial", и т.д.);

generic-family: имя порождающего семейства, ("serif", "sans-serif", "cursive", "fantasy", "monospace").

В качестве разделителя используется запятая. Для последнего варианта желательно определять имя порождающего семейства. Если же имя семейства содержит пробел, то его необходимо поместить в кавычки. При использовании в HTML атрибута "style" применяются одиночные кавычки.

Наследование: да.

Может принимат следующие значения:

Family-name: Определяет для элемента имен семейств шрифтов и/или имен;

Generic-family: порождающих семейств в порядке их приоритета.

Примеры:

h1{font-family: Garamond, serif}

<p style="font-family: Courier, Prestige, sans-serif">

Параметр 'font-size'

Параметр задает размер шрифта.

Наследование: да.

Может принимать следующие значения:

Xx-small x-small small medium large x-large xx-large: Задает размер шрифта от xx-small до xx-large;

Smaller: Задает размер шрифта меньше, чем у порождающего элемента;

Larger: Задает размер шрифта больше, чем у порождающего элемента;

Length: Задает фиксированный размер шрифта;

%: Задает размер шрифта в % от порождающего элемента.

Примеры:

p{font-size: small}

pre{font-size: 5pt}

i{font-size: 3em}

p{font-size: 120%}

Параметр 'font-size-adjust'

Отношение между высотой буквы "x" нижнего регистра в шрифте и высотой "font-size" называется значением аспекта шрифта. Если значение аспекта большое, то шрифт будет легче читаться, когда его размер уменьшается. Например, Verdana имеет значение аспекта, равное 0.58, т.е. если размер шрифта равен 100px, то высота буквы x нижнего регистра будет равна 58px. Times New Roman имеет значение аспекта 0.46. Таким образом, чем больше значение аспекта шрифта, тем легче он читается при уменьшении. В данном случае шрифт Verdana при уменьшении будет читаться лучше, чем Times New Roman.

Свойство font-size-adjust определяет значение аспекта для элемента, которое будет сохранять высоту буквы x нижнего регистра выбранного шрифта.

Наследование: да.

Возможные значения:

None: Не сохранять высоту буквы x шрифта, если шрифт недоступен;

Number: Определяет значение аспекта для шрифта.

Пример:

p{font-size-adjust: none}

p{font-size-adjust: 0.50}

Параметр 'font-stretch'

Данный параметр сжимает или расширяет текущее семейство шрифтов.

Необходимо отметить, что семейство шрифтов сжимается или расширяется только горизонтально.

Наследование: да.

Возможные значения:

Normal: Задает масштаб сжатия или расширения как обычный;

Wider: Задает масштаб расширения как следующее расширенное значение.

Пример:

colspan="2"

p{font-stretch: extra-expanded}

Параметр 'font-style'

Данный параметр задает стиль шрифта.

Наследование: да.

Возможные значения:

Normal: Браузер выводит обычный шрифт;

Italic: Браузер выводит шрифт курсивом;

Oblique: Браузер выводит наклонный шрифт.

Пример:

p{font-style: oblique}

h1{font-style: italic}

Параметр 'font-variant'

Этот параметр используется для вывода текста капителью: все символы нижнего регистра преобразуются в символы верхнего

регистра, при этом все символы шрифта капители имеют меньший размер по сравнению с остальным текстом.

Наследование: да.

Может принимат следующие значения:

Normal: Браузер выводит обычный шрифт;

Small-caps: Браузер выводит шрифт капитель.

Пример:

```
pre{font-variant: small-caps}
p{font-variant: normal}
```

Параметр 'font-weight'

Данный параметр определяет степень жирности символов в тексте при выводе.

Наследование: да.

Возможные значения:

Normal: Определяет обычные символы;

Bold: Определяет жирные символы;

Bolder: Определяет более жирные символы;

Lighter: Определяет более тонкие символы;

100, 200, 300, 400, 500, 600, 700, 800, 900: Определяет степень жирности символов. 400 соответствует normal, 700 соответствует bold

Пример:

```
h1{font-weight: 200}
h2{font-weight: bold}
```

Тема 3.4. Границы в CSS. Отступы в CSS. Поля в CSS.

Цели:

1. *Рассмотреть параметры границ в CSS и способы описания рамок вокруг элементов в CSS.*
2. *Определить методы и свойства, необходимые для описания пространства вокруг элементов.*
3. *Привести способы определения пространства между границей элемента и его содержимым.*

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

Параметры границы в CSS позволяют определить стиль и цвет рамки вокруг элемента. Если в HTML для создания границ вокруг текста использовались таблицы, то в CSS можно создавать границы, которые применимы к любому элементу. Кроме того, возможности параметров границ в CSS расширены.

Параметр 'border-color'

Данный параметр задает цвет четырех сторон границы. Допускается определение от одного до четырех цветов. Необходимо объявлять свойство border-style перед свойством border-color, т.к. элемент должен иметь границу, прежде чем для нее будет задан цвет.

Наследование: нет.

Возможные значения:

Color: Значение цвета может быть названием цвета (red), значением rgb (rgb(255,0,0)) или шестнадцатеричным числом (#FF0000);

Transparent: Граница является прозрачной

Примеры:

p {border-color: blue}

все четыре стороны границы будут синие.

p {border-color: blue yellow}

верхняя и нижняя стороны границы будут синие,

левая и правая стороны границы будут желтые.

p {border-color: blue yellow green}

верхняя стороны граница будет синей,

левая и правая стороны границы будут желтые,

нижняя сторона границы будет зеленая.

p {border-color: blue yellow green red}

верхняя сторона границы будет синей,

правая сторона границы будет желтой,

нижняя сторона границы будет зеленой,

левая сторона границы будет красной.

Параметр 'border-style'

Этот параметр задает стиль четырех сторон границы. Позволяет определить от одного до четырех стилей.

Наследование: нет.

Может принимать следующие значения:

None: Определяет отсутствие границы;

Hidden: То же самое, что "none", за исключением разрешения конфликта границ для элементов таблицы;

Dotted: Определяет пунктирную границу. Изображается в большинстве браузеров сплошной линией;

Dashed: Определяет штрих-пунктирную границу. Изображается в большинстве браузеров сплошной линией;

Solid: Определяет сплошную границу;

Double: Определяет двойную границу. Толщина двойной границы будет соответствовать значению border-width;

Groove: Определяет 3D-границу (groove). Результат зависит от значения цвета границы;

Ridge: Определяет 3D-границу (ridge). Результат зависит от значения цвета границы;

Inset: Определяет 3D-границу (inset). Результат зависит от значения цвета границы;

Outset: Определяет 3D-границу (outset). Результат зависит от значения цвета границы

Примеры:

p {border-style: dashed}

все четыре стороны границы будут штрих-пунктирными.

p {border-style: solid double}

верхняя и нижняя стороны границы будут сплошными,

левая и правая сторона граница будут двойными.

p {border-style: solid double dotted}

верхняя сторона границы будет сплошной,

левая и правая сторона границы будут двойными,

нижняя сторона границы будет пунктирной.

p {border-style: dashed double dotted solid}

верхняя сторона граница будет штрих-пунктирной,

правая сторона границы будет двойной,

нижняя сторона границы будет пунктирной,

левая сторона границы будет сплошной.

Параметр 'border-width'

Данный параметр определяет в одном объявлении толщину четырех сторон границы и может иметь от одного до четырех значений.

Наследование: нет.

Возможные значения:

Thin: Определяет тонкую границу;

Medium: Определяет границы средней толщины;

Thick: Определяет толстую границу;

Length: Позволяет определить толщину границ.

Примеры:

p {border-width: thick}

все четыре стороны границы будут толстыми.

p {border-width: thick thin}

верхняя и нижняя стороны границы будут толстыми,

левая и правая сторона границы будут тонкими.

p {border-width: medium thick thin}

верхняя сторона границы будет иметь среднее значение,

левая и правая стороны границы будут толстыми,

нижняя сторона границы будет тонкой.

p {border-width: thick thin thin medium}

верхняя сторона границы будет толстой,

правая сторона границы будет тонкой,

нижняя сторона границы будет тонкой,

левая сторона границы будет иметь среднее значение.

Отступы в CSS

Они определяют пространство вокруг элементов. Допускают использование отрицательных значений для того, чтобы создавать наложение содержимого. Все отступы (верхний, правый, нижний и левый) можно изменять независимо, используя отдельные параметры, а можно использовать параметр margin для изменения всех отступов одновременно.

Параметр 'margin'

Параметр определяет все отступы в одном объявлении. Позволяет определять отрицательные значения.

Наследование: нет.

Может принимать следующие значения:

Margin-top, margin-right, margin-bottom, margin-left: Задают свойства отступов. Значения задаются в % (определяет отступ в % общей высоты/ширины документа), длиной (определяет фиксированный отступ) и автоматически (отступ задает браузер).

Примеры:

p {margin: 2px}

все четыре отступа будут иметь отступ, равный 2px.

p {margin: 5px 10%}

верхний и нижний отступ будут по 5px,

левый и правый отступ будут составлять 10% общей ширины документа.

p {margin: 5px 10% -2px}

верхний отступ будет равен 5px,

левый и правый отступ будут составлять 10% общей ширины документа, нижний отступ будет равен 2px.

p {margin: 5px 10% -2px auto}

верхний отступ будет 5px,

правый отступ будет составлять 10% общей ширины документа, нижний отступ будет 2px, левый отступ будет задан браузером.

p{ margin-left: 2%}

p{ margin-left: -5px}

p{ margin-right: 2px}

p{ margin-right: 10%}

p{ margin-top: auto}

p{ margin-top: -5px}.

Параметры полей в CSS

Определяют пространство между границей элемента и содержимым элемента, для которого не предусмотрены отрицательные значения. Все поля (верхнее, правое, нижнее и левое) можно изменять независимо друг от друга. Существует также параметр padding, который позволяет определять нескольких полей в одном объявлении.

Параметр 'padding'

Данный параметр предназначен для задания всех полей в одном объявлении. Не позволяет задавать отрицательные значения.

Наследование: нет.

Может принимать следующие значения:

Padding-top, padding-right, padding-bottom, padding-left:

Устанавливает поля. Значения задаются в % (определяет поле в % от ширины ближайшего элемента) и длиной (определяет фиксированное поле).

Примеры:

p {padding: 2px}

для всех четырех сторон будет задано поле 2px.

p {padding: 5% 2px}

верхнее и нижнее поле будет 5%, левое и правое поле будет составлять 2px от ширины ближайшего элемента.

p {padding: 2px 7px 5%}

верхнее поле будет 2px, левое и правое поле будет составлять 7px от ширины ближайшего элемента, нижнее поле будет 5%.

p {padding: 2px 5% 7px 5px}

верхнее поле будет 2px, правое поле будет 5% от ширины ближайшего элемента, нижнее поле будет 7px, левое поле будет 5px.

li{ padding-bottom: 5%}
li{ padding-left: 2px}
li{ padding-left: 5%}
li{ padding-right: 2px}
li{ padding-top: 2px}.

Тема 3.5. Размеры элементов в CSS

Цели:

Рассмотреть параметры размеров элементов и параметры интервалов между элементами. .

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения
Размеры элементов в CSS

Данные параметры CSS позволяют управлять высотой и шириной элемента, увеличением интервала между двумя строками.

Параметр 'height'

Используется для задания высоты элемента.

Наследование: нет.

Пример:

p{height: 50px}

Может принимать следующие значения:

Auto: Браузер вычисляет реальную высоту;

Length: Определяет высоту в px, см, и т.д;

%: Определяет высоту в % от объемлющего блока.

Параметр 'line-height'.

Используется для задания интервала между строками.

Наследование: Да.

Примечание: Отрицательные значения не допускаются.

Примеры:

div{ line-height: 2.1}

div{ line-height: 10pt}

div{ line-height: 120%}

Может принимать следующие значения:

Normal: Задает приемлемый интервал между строками;

Number: Задает число, которое при умножении на размер текущего шрифта задает интервал между строками;

Length: Задает фиксированный интервал между строками;

%: Задает интервал между строками в % от размера текущего шрифта.

Параметр 'max-height'

Используется для задания максимальной высоты элемента.

Наследование: нет.

Пример:

div{max-height: 200%}

Может принимать следующие значения:

None: Отсутствие ограничений на максимальную высоту элемента;

Length: Определяет максимальную высоту элемента;

%: Определяет максимальную высоту элемента в % от объемлющего блока.

Параметр 'max-width'

Используется для задания максимальной ширины элемента.

Наследование: нет.

Пример:

h6{max-width: 75%}

Может принимать следующие значения:

None: Отсутствие ограничений на максимальную ширину элемента;

Length: Определяет максимальную ширину элемента;

%: Определяет максимальную ширину элемента в % от объемлющего блока.

Параметр 'min-height'

Используется для задания минимальной высоты элемента.

Наследование: нет.

Пример:

div{min-height: 20%}

Может принимать следующие значения:

Length: Определяет минимальную высоту элемента;

%: Определяет минимальную высоту элемента в % от объемлющего блока.

Параметр 'min-width'

Используется для задания минимальной ширины элемента.

Наследование: нет.

Пример:

h6{min-width: 20%}

Может принимать следующие значения:

Length: Определяет минимальную ширину элемента;

%: Определяет минимальную ширину элемента в % от объемлющего блока.

Параметр 'width'

Используется для задания ширины элемента.

Наследование: нет.

Пример:

hr{width: 20%}

Может принимать следующие значения:

Auto: Браузер вычисляет реальную ширину;

%: Определяет ширину в % от ширины родительского элемента;

Length: Определяет ширину в px, см, и т.д.

Тема 3.6. Параметры форматирования в CSS

Цели:

Определить способы управления видимостью, положением и позиционированием элементов.

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

Данные параметры позволяют: управлять выводом элемента; задавать положение изображения в другом элементе; позиционировать элемент относительно его обычного положения; позиционировать

элемент с помощью абсолютного значения; управлять видимостью элемента.

Параметр 'clear'

Данный параметр задает стороны элемента, где не допускаются другие плавающие элементы. Плавающие элементы - изображения и текстовые элементы, которые появляются в другом элементе.

Примечание: При использовании с параметром "float" не всегда работает как ожидается.

Наследование: нет.

Пример:

div{clear: both}

div{clear: left}

Может принимать следующие значения:

Left: Плавающие элементы не допускаются слева;

Right: Плавающие элементы не допускаются справа;

Both: Плавающие элементы не допускаются ни справа, ни слева;

None: Плавающие элементы допускаются с любой стороны.

Параметр 'cursor'

Данный параметр определяет тип курсора, который будет выводиться при указании на элемент.

Наследование: да.

Примеры:

pre{cursor: pointer}

div{cursor : url("general.cur"), url("other.cur"), text;}

Может принимать следующие значения:

url: url специального курсора, который будет использоваться;

default: Курсор по умолчанию (часто стрелка);

auto: Курсор задает браузер;

crosshair: Курсор изображается как перекрестие;

pointer: Курсор изображается как указатель (рука), который обозначает ссылку;

move: Курсор указывает объект, который можно переместить;

e-resize: Курсор, определяющий перемещение края бокса вправо (east);

ne-resize: Курсор, определяющий перемещение края бокса вверх и вправо (north/east);

nw-resize: Курсор, определяющий перемещение края бокса вверх и влево (north/west);

n-resize: Курсор, определяющий перемещение края бокса вверх (north);

se-resize: Курсор, определяющий перемещение края бокса вниз и вправо (south/east);

sw-resize: Курсор, определяющий перемещение края бокса вниз и влево (south/west);

s-resize: Курсор, определяющий перемещение края бокса вниз (south);

w-resize: Курсор, определяющий перемещение края бокса влево (west);

text: Курсор, используемый для текста;

wait: Курсор, указывающий на занятость программы (часто песочные часы);

help: Курсор, указывающий, что имеется справочная информация (часто знак вопроса).

Параметр 'display'

Данный параметр определяет, как в документе будет показан элемент.

Наследование: нет.

Примеры:

pre{display: block}

strong{display: inline}

img{display: none}

h1, h3{display: no}

Может принимать следующие значения:

None: Элемент не будет выводиться;

Inline: Элемент будет выводиться как элемент уровня блока с разрывом строки перед и после элемента;

Block: Элемент будет выводиться как элемент строки без разрыва строки перед или после элемента;

list-item: Элемент будет выводиться как список;

run-in: Элемент будет выводиться на уровне блока или как внутристрочный элемент в зависимости от контекста;

compact: Элемент будет выводиться на уровне блока или как внутристрочный элемент в зависимости от контекста;

table: Элемент будет выведен как таблица блока (как <table>), с разрывом строки перед и после таблицы;

inline-table: Элемент будет выведен как внутристрочная таблица (как <table>), без разрыва строки перед или после таблицы;

table-row-group: Элемент выводится как группа из одной или нескольких строк (как <tbody >);

table-header-group: Элемент выводится как группа из одной или нескольких строк (как <thead>).

Свойство 'float'

Данное свойство определяет, где появится в другом элементе изображение или текст.

Примечания:

Если в строке слишком мало места для плавающего элемента, он переносится на следующую строку и продолжается, пока строка имеет достаточно места.

Перед float должны идти - содержание, фон, и границы внутристрочных элементов, содержание блочного элемента, после - фон и границы блочного элемента.

Наследование: нет.

Пример:

p{float: right}

Может принимать следующие значения:

Left: Изображение или текст смещается в родительском элементе влево;

Right: Изображение или текст смещается в родительском элементе вправо;

None: Изображение или текст будут выводиться в том месте, где они встретились в тексте;

Параметр 'position'

Данный параметр помещает элемент в статическое, относительное, абсолютное или фиксированное положение.

Наследование: нет.

Пример:

p{position:static;}

Может принимать следующие значения:

Static: Элемент помещается в обычное положение (согласно нормальному потоку). Со значением "static" не используются параметры "left" и "top";

Relative: Смещает элемент относительно его нормального положения, поэтому "left:20" добавляет 20 пикселей к позиции LEFT элемента;

Absolute: С помощью значения "absolute" элемент можно поместить в любом месте страницы. Позиция элемента определяется с помощью параметров "left", "top", "right", и "bottom";

fixed

Параметр 'visibility'

Этот параметр определяет видимость или невидимость элемента.

Примечания:

Невидимые элементы занимают место на странице. Для того, чтобы создать невидимые элементы, которые не занимают место, необходимо использовать параметр "display".

Этот параметр можно использовать со сценариями для создания Динамического HTML.

Наследование: нет.

Примеры:

pre{visibility: hidden}

Может принимать следующие значения:

Visible: Элемент видим;

Hidden: Элемент невидим;

Collapse: При использовании в табличных элементах это значение удаляет строку или столбец, но не влияет на компоновку таблицы. Пространство, занимаемое строкой или столбцом, будет

доступно для другого содержимого. Если этот параметр используется с другими элементами, он действует как "hidden".

Тема 3.7. Позиционирование в CSS

Цели:

Рассмотреть методы определения положения элемента..

Рекомендуемая литература:

1. Пятибратов, А.П. Вычислительные системы, сети и телекоммуникации : учебник для студ. вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; ред. А. П. Пятибратов. - 4-е изд., перераб. и доп. - М. : ИНФРА-М, 2014. - 734 с.
2. Храмцов П.Б., Брик С.А., Русак А.М., Сурин А.И. Основы web-технологий / БИНОМ. Лаборатория знаний, интернет-университет информационных технологий - ИНТУИТ.ру, 2009. – 376 с.
3. http://htmlbook.ru/layout – Веб-технологии
4. http://habrahabr.ru/blogs/css/126207/ – Разработка таблиц стилей

Теоретические сведения

Параметры позиционирования в CSS позволяют: определить левую, правую, верхнюю и нижнюю позиции элемента; задать форму элемента; поместить элемент позади другого; определить, что будет происходить, когда содержимое элемента слишком большое для размещения в указанной области.

Параметр 'bottom'

Данный параметр определяет нижний край элемента.

Примечание: Если параметр "position" имеет значение "static", параметр "bottom" не оказывает влияния.

Наследование: нет.

Примеры:

В примере показано, как задать нижний край элемента pre на 50 px выше нижнего края окна:

```
pre{position: absolute;
bottom: 50px}
```

В примере показано, как задать нижний край элемента pre на 50 px ниже нижнего края окна:

pre{position: absolute;

bottom: -50px}

Может принимать следующие значения:

Auto: Позволяет браузеру самостоятельно вычислить нижнюю позицию;

%: Задает нижнюю позицию в % от положения нижнего края окна;

Length: Задает нижнюю позицию в px, см, и т.д. от нижнего края окна. Допускаются отрицательные значения.

Параметр 'clip'

Данный параметр задает форму элемента.

В ситуации если, например, изображение больше того элемента, в котором располагается, параметр позволяет определить размеры изображения, обрезать по форме и вывести.

Примечание: Этот параметр нельзя использовать для элементов с параметром "overflow", заданным как "visible".

Наследование: нет.

Пример:

p{position:absolute;

clip:rect(2px 175px 100px 0px)}

Может принимать следующие значения:

Shape: Задает форму элемента. Допустимым значением формы является: rect (top, right, bottom, left);

Auto: Браузер задает форму элемента.

Параметр 'left'

Данный параметр определяет левый край элемента.

Примечание: Если параметр "position" имеет значение "static", параметр "left" не оказывает влияния.

Наследование: нет.

Примеры:

В примере показано, как задать левый край элемента pre в 50 px справа от левого края окна:

pre{position: absolute;

left: 50px}

В примере показано, как задать левый край элемента pre в 50 px слева от левого края окна:

pre{position: absolute;

left: -50px}

Может принимать следующие значения:

Auto: Позволяет браузеру вычислить левую позицию;

%: Задает левую позицию в % от значения для левого края окна;

Length: Задает левую позицию в px, см и т.д. от левого края окна. Допускаются отрицательные значения.

Параметр 'overflow'

Данный параметр определяет, что происходит, когда содержимое элемента переполняет его область.

Наследование: нет.

Пример:

div {overflow: auto}

Может принимать следующие значения:

Visible: Содержимое не обрезается. Оно выводится за пределами элемента;

Hidden: Содержимое обрезается, но браузер не выводит полосу прокрутки для просмотра всего содержимого;

Scroll: Содержимое обрезается, но браузер выводит полосу прокрутки для просмотра всего содержимого;

Auto: Если содержимое обрезается, то браузер должен вывести полосу прокрутки для просмотра всего содержимого.

Параметр 'position'

Данный параметр помещает элемент в статическое, относительное, абсолютное или фиксированное положение.

Наследование: нет.

Пример:

p{position:static;}

Может принимать следующие значения:

Static: Элемент помещается в обычное положение (согласно нормальному потоку). Для значения "static" параметры "left" и "top" не используются;

Relative: Перемещает элемент относительно нормального положения, так что "left:20" добавляет 20 пикселей к позиции LEFT элемента;

Absolute: С помощью значения "absolute" элемент можно разместить в любом месте страницы. Позиция элемента определяется параметрами "left", "top", "right", и "bottom";

Fixed.

Параметр 'right'

Данный параметр определяет правый край элемента.

Примечание: Если параметр "position" имеет значение "static", то параметр "right" не оказывает влияния.

Наследование: нет.

Примеры:

В примере показано, как задать правый край элемента pre на 50 px влево от правого края окна:

```
pre{position: absolute;
right: 50px}
```

В примере показано, как задать правое поле элемента pre на 50 px вправо от правого края окна:

```
pre{position: absolute;
right: -50px}
```

Может принимать следующие значения:

Auto: Позволяет браузеру вычислить правую позицию;

%: Задает правую позицию в % от значения правого края окна;

Length: Задает правую позицию в px, см, и т.д. от правого края окна. Допускаются отрицательные значения.

Параметр 'top'

Данный параметр определяет верхний край элемента.

Примечание: Если параметр "position" имеет значение "static", то параметр "top" не оказывает влияния.

Наследование: нет.

Примеры:

На примере показано, как задать верхний край элемента pre на 50 px ниже верхнего края окна:

pre{position: absolute;

top: 50px}

На примере показано, как задать верхний край элемента pre на 50 px выше верхнего края окна:

pre{position: absolute;

top: -50px}

Может принимать следующие значения:

Auto: Позволяет браузеру вычислить верхнюю позицию;

%: Задает верхнюю позицию в % от значения верхнего края окна;

Length: Задает верхнюю позицию в px, см, и т.д. от верхнего края окна. Допускаются отрицательные значения.

Параметр 'vertical-align'

Данный параметр задает вертикальное выравнивание элемента.

Наследование: нет.

Примеры:

img{ vertical-align: baseline}

Может принимать следующие значения:

Baseline: Элемент размещается на базовой строке родительского элемента;

Sub: Выравнивает элемент как нижний индекс;

Super: Выраванивает элемент как верхний индекс;

Top: Вершина элемента выравнивается с вершиной самого высокого элемента в строке;

text-top: Вершина элемента выравнивается с вершиной шрифта родительского элемента;

middle: Элемент помещается в середине родительского элемента;

bottom: Нижняя часть элемента выравнивается с самым нижним элементом в строке;

text-bottom: Нижняя часть элемента выравнивается с минимальной нижней точкой родительского элемента;

length

%: Выравнивает элемент в % от значения параметра "line-height". Допускаются отрицательные значения.

Параметр 'z-index'

Данный параметр задает порядковый номер элемента в стеке. Элемент с большим порядковым номером стека всегда находится перед элементом с меньшим порядковым номером стека.

Примечания:

Элементы могут иметь отрицательные порядковые номера стека.

Z-index работает только с теми элементами, которые были позиционированы (например, position:absolute;)!

Наследование: нет.

Пример:

```
img{position:absolute
z-index: 1}
```

Может принимать следующие значения:

Auto: Порядковый номер элемента в стека равен номеру родительского элемента

Практическая работа №1. Работа с каскадными таблицами стилей

Цели:

1. *научиться работать с каскадными таблицами стилей;*
2. *изучить основные методы работы с CSS, селекторы, атрибуты.*

Задание

Оформить подготовленный структурированный гипертекст, представленный в файле index.html в соответствии со стилем представленным на рисунке index.bmp.

При выполнении работы допускается использовать «песочницу» http://cssdesk.com/ и справочник http://htmlbook.ru/css.

Для подробного освоения темы по «margin» и «padding» рекомендуется прочитать:
http://habrahabr.ru/blogs/css/121810/
http://habrahabr.ru/blogs/css/123250/

Исходный HTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html><head><title>Template</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1" />
<style type="text/css" media="all">
@import "style.css";
</style></head>
<body>
<div class="content">
  <div class="toph"></div>
  <div class="right">
    <div class="title">GREY</div>
    <div class="nav">      <ul>
        <li><a href=#>HOME</a></li>
        <li><a href=#>ARTICLES</a></li>
        <li><a href=#>GALLERY</a></li>
        <li><a href=#>AFFILIATES</a></li>
        <li><a href=#>SUPPORT</a></li>
        <li><a href=#>CONTACT</a></li>
      </ul>    </div>
    <h2>Top Articles:</h2>
    <ul>
      <li><a href=#>NoHeader Template</a></li>
      <li><a href=#>Consectetuer adipiscing elit</a></li>
      <li><a href=#>Lorem ipsum dolor sit amet</a></li>
      <li><a href=#>dolor sit amet consectet</a></li>
    </ul>
    <hr />
    <h2>Links</h2>
    <ul>
      <li><a href=#>any.com</a></li>
      <li><a href="htmlbook.ru/samcss">htmlbook.ru/samcss</a></li>
    </ul>    <hr />  </div>  <div class="center">
    <h2><a href=#>Try sNews 1.4!</a></h2>
    Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Integer euismod
ante non diam. Sed eleifend odio sed quam. Sed vulputate, <a href=#>turpis
```

at tincidunt porttitor, est elit consequat metus, non dignissim augue mauris quis arcu. Phasellus faucibus blandit eros. Curabitur porttitor ante non est. Maecenas dolor. Aenean egestas sem. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Sed suscipit, nisi sit amet pharetra malesuada, sem velit laoreet sem, vitae iaculis diam neque consequat est. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Pellentesque tincidunt eros non quam. Mauris a magna sit amet libero accumsan auctor. Aenean nec urna non dui lobortis viverra...

```
    <p class="date">Posted by Avenir <img src="more.gif" alt="" />
 <a href=#>Read more</a> <img src="comment.gif" alt="" />
<a href=#>Comments (2)</a> <img src="timeicon.gif" alt="" />
21.02.</p>
    <br />
    <h2><a href=#>Heading Item</a></h2>
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Integer euismod ante non diam. Sed eleifend odio sed quam. Sed vulputate, turpis at tincidunt porttitor, est elit consequat metus, non dignissim augue mauris quis arcu. Phasellus faucibus blandit eros.

```
    <p class="date">Posted by James <img src="more.gif" alt="" />
<a href=#>Read more</a> <img src="comment.gif" alt="" />
 <a href=#>Comments (7)</a> <img src="timeicon.gif" alt="" />
18.01.</p>
    <br />
    <div class="boxad"> Your Ads here...sNews is a completly free PHP
and MySQL driven tool for publishing and maintaining news articles on a
website.</div>
  </div>
  <div class="footer"></div>
</div>
</body>
</html>
```

Требуемый стиль:



Ход работы

1. Для селектора «body»:

- установить цвет фона тела страницы #7D8085, используя свойство «background» селектора «body»;
- установить шрифт тела страницы 74% Arial, Sans-Serif, используя свойство «font»;

2. Для селектора (класса) «.toph»:

- установить неповторяющийся фоновый рисунок «top.jpg» без полей, высотой 39px, с выравниванием по центру, используя свойства «background», «height», «margin», «padding»;

3. Для селектора (класса) «.content»:

- установить повторяющийся фоновый рисунок «bg.jpg» без полей, шириной 800px, с выравниванием по центру, используя свойства «background», «width», «margin», «padding»;

4. Для селектора (класса) «.title»:

- установить неповторяющийся фоновый рисунок «logo.jpg» с выравниванием по левой стороне, используя свойство «background»;
- с полями от верхнего края 10px, от левого 40px, высотой 28px, используя свойства «height», «padding»;
- установить размер шрифта 140%, полужирный, цвет #F29900, используя свойство «font»;

5. Для селектора (класса) «.right»:
- установить обтекание слева, используя свойство «float»;
- с отступом от правого края 15px, и полем от правого края 1em, используя свойства «margin», «padding»;
- установить размер шрифта 95%, полужирный, используя свойство «font»;
- установить ширину слоя 170px, используя свойство «width»;
6. Для селектора (класса) «.footer»:
- установить запрет на обтекание одновременно с правого и левого края, используя свойство «clear»;
- установить неповторяющийся фоновый рисунок «bot.jpg» с выравниванием по центру стороне, используя свойство «background»;
- установить выравнивание текста по центру, используя свойство «text-align»;
- установить высоту слоя в 37px, используя свойство «height»;
- установить автоматическую ширину слоя «auto», используя свойство «width»;
7. Для селектора (класса) «.center»:
- установить обтекание справа, используя свойство «float»;
- установить ширину слоя 530px, используя свойство «width»;
- установить размер шрифта 95%, полужирный, используя свойство «font»;
- установить цвет текста #FFF;
- установить поля и отступы соответственно «margin: 0px 0 5px 35px; padding: 0;»;
8. Установить цвет ссылок для центрального блока:
- цвет основной ссылки #F29900, используя селекторы «.center a»;
- цвет ссылки под курсором #FFF, используя селекторы «.center a:hover»;
9. Установить для блока «date»:
- цвет основного текста #ccc
- выравнивание текста по правому краю, используя свойство «text»
- поля и отступы соответственно «margin: 4px 0 5px 0; padding: 0.4em 0 0 0;»
- верхнюю границу блока толщиной в 1px цветом #555, используя свойство «border»;
10. Установить цвет ссылки в #ccc, используя свойство «color» селектора «.date a»;

11. Установить цвет ссылки в #7D8085, используя свойство «color» селектора «.right a»;
12. Установить цвет тегов параграфа и ссылок в #888, используя свойство «color» селекторов «p» и «a»;
13. Для селектора «a»:
- установить наследование фона, используя свойство «background» с параметром «inherit»;
- выключить стили текста, используя свойство «text-decoration»;
14. Для селектора «p» установить отступы и поля соответственно «margin: 0 0 5px 0; padding: 0;»
15. Для селектора «hr»:
- установить высоту 1px;
- установить основной и фоновый цвет в #eee;
- убрать «border»;
16. Для селектора заголовка «h1»:
- убрать отступы и поля;
- установить цвет в #FFF;
- установить жирный шрифт размера 1.8em, гарнитур Arial, Sans-Serif;
- установить наследование фона, используя свойство «background» с параметром «inherit»;
-установить свойство letter-spacing равным «-1px»;
17. Установить цвет для ссылок «a», находящихся в заголовке «h1» в #FFF, и установить для них наследование фона.
18. Для селектора заголовка «h2»:
- установить наследование цвета фона, используя свойство «background» с параметром «inherit»;
- установить отступы и поля согласно «margin: 10px 0 10px 0; padding:0;»
- установить основной цвет в #F29900;
- установить размер шрифта 140%, жирный;
19. Установить цвет для ссылок «a» находящихся в заголовке «h2» в #F29955. Для ссылок под курсором установить тот же цвет и убрать подчёркивание.
20. Применить стили элементов списка согласно инструкциям и вставить комментарии действия свойств:

```
ul { margin: 5px 0 20px 15px;
     padding: 0;
     list-style: none;
}
```

```
li { list-style-type: square;
    color: #F29900;
    margin: 0 0 0px 0;
    padding: 0 0 0 0px;
}
li a { color: #7D8085; }
li a:hover { color: #F29900;  }
```

21. Выполнить применение указанных стилей тремя методами: связанным, глобальным и внутренним

22. **Дополнительно**:
- рассмотреть работу базовых макетов построения сайта представленных в http://htmlbook.ru/layout
- ознакомиться с материалами статьи http://habrahabr.ru/blogs/css/126207/.

Практическая работа №2. Экспорт стилей и валидация.

Цели:

1. *научиться экспортировать стили;*
2. *научиться структурировать документ.*

Ход работы

Используя материал предыдущей работы, выполнить следующее задание:

1. Сменить заголовок с «GRAY» на «CSS», подобрать логотип.
2. Перенести из учебника материал, по числу пунктов меню начиная с введения.
3. Выполнить хотя бы одно абсолютное и относительное позиционирование (http://ru.html.net/tutorials/css/lesson14.php).
4. Сделать хотя бы один плавающий блок – картинку (см. http://ru.html.net/tutorials/css/lesson13.php).
5. Сделать наслоение (http://ru.html.net/tutorials/css/lesson15.php).
6. Указать у всех используемых шрифтов родовые имена и объяснить, зачем это нужно. (http://ru.html.net/tutorials/css/lesson4.php).
7. Выполнить экспорт стилей в файл и осуществить его подключение.

8. Выполнить валидацию файла стиля (см. http://ru.html.net/tutorials/css/lesson16.php)

Литература

1. Бен Хеник, HTML и CSS. Путь к совершенству. СПб: Питер, 2011 -336 с.
2. Дунаев В., HTML, скрипты и стили. СПб: БХВ-Петербург, 2011-816 с.
3. Дронов В., HTML 5, CSS 3 и Web 2.0. Разработка современных web-сайтов. СПб: БХВ-Петербург, 2011 - 416 с.
4. Джон Поллок. JavaScript. Руководство разработчика. СПб: Питер,2011-544 с.
5. Дэвид Макфарланд. JavaScript. Подробное руководство. Эксмо, 2009-608 с.
6. Чак Муссиано, Билл Кеннеди. HTML и XHTML. Подробное руководство. Символ-Плюс, 2011 - 752 с.
7. Кисленко Н.П. HTML. Самое необходимое. СПб: БХВ-Петербург, 2012 – 352 с.
8. Комолова Н., Яковлева Е. HTML, XHTML и CSS. СПб: Питер, 2012 –304 с.
9. Пол Вилтон, Джереми МакПик. JavaScript. Руководство программиста. СПб: Питер, 2009-720 с.
10. Роберт Агулар. HTML и CSS. Основа любого сайта. Экспо, 2010 –320 с.
11. Стоян Стефанов. JavaScript. Шаблоны. СПб: Символ-плюс, 2011-272с.
12. Климов А. JavaScript на примерах. СПб: БХВ-Петербург, 2009-336 с.
13. Шафер С., HTML, XHTMLи CSS. Библия пользователя. М.: Вильямс, 2010 – 656 с.
14. Лабберс К., Олберс Н., Салим К.. HTML5 для профессионалов:мощные инструменты для разработки современных веб-приложений. М.: Вильямс, 2011 – 272 с.

Интернет-ресурсы

1. www.w3.org
2. www.intuit.ru/studies/courses. - Открытые интернет-курсы «Интуит» по курсу «Информатика»

3. lib.kchgu.ru. - Электронная библиотека КЧГУ | ФГБОУ ВО КЧГУ имени У.Д. Алиева
4. http://javascript.itsoft.ru -Online-справочник по языку JavaScript
5. http://doks.gorodok.net/ - Справочники и учебники по PHP, C/C++, HTML, CSS, Javascript, XML, SQL, Perl, Windows, Unix
6. http://computerlibrary.info/ - Рубрики: Веб-мастеру, интернет, Софт
7. http://www.webclub.ru/ - WebClub - Всероссийский Клуб Веб-разработчиков
8. http://dweb.ru/rass/html/index.htm - Учебник по HTML
9. http://www.htmlbook.ru/ - посвящен языку HTML, CSS, веб-дизайну, графике и процессу создания сайтов
10. http://www.codenet.ru - все для программиста
11. http://html.krsk.ru/css/index.asp - Каскадные таблицы стилей

Приложение 1.

Цвета в HTML

Цвета выводятся с помощью смешения источников:

RED (красного) цвета:

GREEN (зеленого) цвета:

BLUE (синего) цвета.

Значения цветов

Цвета в HTML определяют с помощью шестнадцатеричной записи комбинации значений красного, зеленого и синего цветов (RGB). Наименьшее значение, которое можно задать одному из источников равно 0 (hex #00), а максимальное - 255 (hex #FF).

Следующая таблица показывает результат объединения источников красного (R), зеленого (G) и синего (B) источников цвета:

Цвет	Цвет (HEX)	Цвет RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

Названия цветов

Некоторая совокупность названий цветов поддерживается большинством браузеров.

Примечание: Только 16 названий цветов поддерживается стандартом W3C для CSS (aqua (голубой), black (черный), blue (синий), fuchsia (фуксия), gray (серый), green (зеленый), lime (лайм), maroon (темно-бордовый), navy (темно-синий), olive (оливковый), purple (сиреневый), red (красный), silver (светло-серый), teal (сине-зеленый), white (белый), и yellow (желтый)). Для всех других цветов необходимо использовать значение HEX цвета.

Цвет	Цвет (HEX)	Название цвета

	#FFF8DC	Cornsilk (кукурузные рыльца)
	#00008B	DarkBlue (темно-синий)
	#008B8B	DarkCyan (темно-голубой)
	#FFFFF0	Ivory (слоновая кость)
	#FFFFE0	LightYellow (светло-желтый)
	#FFA500	Orange (оранжевый)
	#4169E1	RoyalBlue (ярко-синий)

Названия цветов в HTML

На этой странице находится таблица названий цветов, которые поддерживаются большинством броузеров.

Примечание: Стандарт HTML консорциума W3C поддерживает только 16 имен цветов (aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow). Для всех остальных цветов необходимо использовать HEX (шестнадцатеричное) значение цвета.

Название цвета	HEX цвета	Цвет
AliceBlue (бледно-голубой)	#F0F8FF	
AntiqueWhite (античный белый)	#FAEBD7	
Aqua (голубой)	#00FFFF	
Aquamarine (аквамарин)	#7FFFD4	
Azure (лазурный)	#F0FFFF	
Beige (беж)	#F5F5DC	
Bisque (бисквитный)	#FFE4C4	
Black (черный)	#000000	
BlanchedAlmond (светло-кремовый)	#FFEBCD	
Blue (синий)	#0000FF	
BlueViolet (сине-фиолетовый)	#8A2BE2	
Brown (коричневый)	#A52A2A	
BurlyWood (старое дерево)	#DEB887	
CadetBlue (серо-голубой)	#5F9EA0	
Chartreuse (зеленовато-желтый)	#7FFF00	
Chocolate (шоколадный)	#D2691E	
Coral (коралловый)	#FF7F50	
CornflowerBlue (васильковый)	#6495ED	
Cornsilk (кукурузные рыльца)	#FFF8DC	
Crimson (малиновый)	#DC143C	
Cyan (светло-голубой)	#00FFFF	
DarkBlue (темно-синий)	#00008B	

DarkCyan (темно-голубой)	#008B8B	
DarkGoldenRod (темно-золотисто-коричневый)	#B8860B	
DarkGray (темно-серый)	#A9A9A9	
DarkGreen (темно-зеленый)	#006400	
DarkKhaki (темный хаки)	#BDB76B	
DarkMagenta (темно-пурпурный)	#8B008B	
DarkOliveGreen (темно-оливково-зеленый)	#556B2F	
Darkorange (темно-оранжевый)	#FF8C00	
DarkOrchid (темно-лиловый)	#9932CC	
DarkRed (темно-красный)	#8B0000	
DarkSalmon (темно-оранжево-розовый)	#E9967A	
DarkSeaGreen (темно-зеленое море)	#8FBC8F	
DarkSlateBlue (темный серовато-синий)	#483D8B	
DarkSlateGray (темный синевато-серый)	#2F4F4F	
DarkTurquoise (темно-бирюзовый)	#00CED1	
DarkViolet (темно-фиолетовый)	#9400D3	
DeepPink (темно-розовый)	#FF1493	
DeepSkyBlue (темно-небесно-голубой)	#00BFFF	
DimGray (тускло-серый)	#696969	
DodgerBlue (тускло-васильковый)	#1E90FF	
Feldspar (полевой шпат)	#D19275	
FireBrick (огнеупорный кирпич)	#B22222	
FloralWhite (цветочно-белый)	#FFFAF0	
ForestGreen (лесная зелень)	#228B22	
Fuchsia (фуксия)	#FF00FF	
Gainsboro (светло-серый)	#DCDCDC	
GhostWhite (туманно-белый)	#F8F8FF	
Gold (золотой)	#FFD700	
GoldenRod (золотисто-коричневый)	#DAA520	
Gray (серый)	#808080	
Green (зеленый)	#008000	
GreenYellow (зелено-желтый)	#ADFF2F	
HoneyDew (медовая роса)	#F0FFF0	
HotPink (ярко-розовый)	#FF69B4	

IndianRed (индийский-красный)	#CD5C5C	
Indigo (индиго)	#4B0082	
Ivory (слоновая кость)	#FFFFF0	
Khaki (хаки)	#F0E68C	
Lavender (бледно-лиловый)	#E6E6FA	
LavenderBlush (бледный розово-лиловый)	#FFF0F5	
LawnGreen (зеленая трава)	#7CFC00	
LemonChiffon (лимонный)	#FFFACD	
LightBlue (светло-синий)	#ADD8E6	
LightCoral (светло-коралловый)	#F08080	
LightCyan (светло-светло-голубой)	#E0FFFF	
LightGoldenRodYellow (светлый коричнево-желтый)	#FAFAD2	
LightGrey (светло-серый)	#D3D3D3	
LightGreen (светло-зеленый)	#90EE90	
LightPink (светло-розовый)	#FFB6C1	
LightSalmon (светлый оранжево-розовый)	#FFA07A	
LightSeaGreen (светло-зеленое море)	#20B2AA	
LightSkyBlue (светло-небесно-голубой)	#87CEFA	
LightSlateBlue (светлый серовато-синий)	#8470FF	
LightSlateGray (светлый синевато-серый)	#778899	
LightSteelBlue (светлый голубовато стальной)	#B0C4DE	
LightYellow (светло-желтый)	#FFFFE0	
Lime (лайм)	#00FF00	
LimeGreen (зеленый лайм)	#32CD32	
Linen (льняной)	#FAF0E6	
Magenta (пурпурный)	#FF00FF	
Maroon (каштановый)	#800000	
MediumAquaMarine (умеренно аквамариновый)	#66CDAA	
MediumBlue (умеренно синий)	#0000CD	
MediumOrchid (умеренно лиловый)	#BA55D3	
MediumPurple (умеренно пурпурный)	#9370D8	

MediumSeaGreen (умеренно-зеленое море)	#3CB371	
MediumSlateBlue (умеренно серовато-синий)	#7B68EE	
MediumSpringGreen (умеренная весенняя зелень)	#00FA9A	
MediumTurquoise (умеренно-бирюзовый)	#48D1CC	
MediumVioletRed (умеренно фиолетово красный)	#C71585	
MidnightBlue (ночной синий)	#191970	
MintCream (мятно-кремовый)	#F5FFFA	
MistyRose (туманно-розовый)	#FFE4E1	
Moccasin (песочный)	#FFE4B5	
NavajoWhite (темно-песочный)	#FFDEAD	
Navy (темно-синий)	#000080	
OldLace (старые кружева)	#FDF5E6	
Olive (оливковый)	#808000	
OliveDrab (тускло-оливковый)	#6B8E23	
Orange (оранжевый)	#FFA500	
OrangeRed (оранжево-красный)	#FF4500	
Orchid (лиловый)	#DA70D6	
PaleGoldenRod (бледно-золотисто-коричневый)	#EEE8AA	
PaleGreen (бледно-зеленый)	#98FB98	
PaleTurquoise (бледно-голубой)	#AFEEEE	
PaleVioletRed (бледно-фиолетово-красный)	#D87093	
PapayaWhip (дыни)	#FFEFD5	
PeachPuff (персиковый)	#FFDAB9	
Peru (коричневый)	#CD853F	
Pink (розовый)	#FFC0CB	
Plum (сливовый)	#DDA0DD	
PowderBlue (туманно-голубой)	#B0E0E6	
Purple (пурпурный)	#800080	
Red (красный)	#FF0000	
RosyBrown (розово-коричневый)	#BC8F8F	
RoyalBlue (ярко-синий)	#4169E1	
SaddleBrown (старая кожа)	#8B4513	
Salmon (оранжево-розовый)	#FA8072	
SandyBrown (песочно-коричневый)	#F4A460	

SeaGreen (зеленое море)	#2E8B57	
SeaShell (морская ракушка)	#FFF5EE	
Sienna (охра)	#A0522D	
Silver (свето-серый)	#C0C0C0	
SkyBlue (небесно-голубой)	#87CEEB	
SlateBlue (серовато-синий)	#6A5ACD	
SlateGray (синевато-серый)	#708090	
Snow (снежный)	#FFFAFA	
SpringGreen (весенняя зелень)	#00FF7F	
SteelBlue (сине-стальной)	#4682B4	
Tan (бронзы)	#D2B48C	
Teal (сине-зеленый)	#008080	
Thistle (чертополох)	#D8BFD8	
Tomato (томат)	#FF6347	
Turquoise (бирюзовый)	#40E0D0	
Violet (фиолетовый)	#EE82EE	
VioletRed (фиолетово-красный)	#D02090	
Wheat (пшеничный)	#F5DEB3	
White (белый)	#FFFFFF	
WhiteSmoke (белый дым)	#F5F5F5	
Yellow (желтый)	#FFFF00	
YellowGreen (желто-зеленый)	#9ACD32	

Приложение 2.
Раздел 1. Архитектура web-пространства

1. В глобальных сетях существуют два режима информационного обмена э
   a) пользовательский и сетевой;
   b) информируемый и скрытый;
   c) диалоговый и пользовательский;
   d) диалоговый и пакетный.
2. On-line — это:
   a) информационная сеть;
   b) команда;
   c) режим реального времени;
   d) утилита.
3. Крупнейшая российская телекоммуникационная сеть:
   a) BITNET;
   b) APRANET;
   c) NET;
   d) RELCOM.
4. Of-line — это:
   a) режим информационного пакетного обмена;
   b) команда;
   c) телекоммуникационная сеть;
   d) операционная система.
5. Мировая система телеконференций:
   a) Eunet;
   b) Fidonet;
   c) Relcom;
   d) Usenet.
6. Хост-машина — это:
   a) банк информации;
   b) компьютерные узлы связи;
   c) мультимедийный компьютер;

d) машина-хранилище информации.

7. Модем — это:

a) устройство преобразования цифровых сигналов в аналоговые, и наоборот;
b) транспортная основа сети;
c) хранилище информации;
d) устройство, которое управляет процессом передачи информации.

8. Что обеспечивают протоколы сетевого уровня:

a) обеспечивают сетевые режимы передачи данных;
b) доступ к сетевым ресурсам;
c) соединяют различные сети;
d) тестируют работу в сети?

9. Транспортные протоколы выполняют следующие функции:

a) группируют сообщения;
b) кодируют пакеты информации;
c) отвечают за обмен между хост-машинами;
d) контролируют вход и выход данных.

10. За что отвечают прикладные протоколы:

a) за передачу данных и доступ к сетевым ресурсам;
b) формируют пакеты данных;
c) контролируют работу хост-машин;
d) тестируют правильность работы сети?

11. Техническая структура E-mail — это:

a) совокупность узловых станций, связывающихся друг с другом для обмена
b) совокупность компьютеров локальной сети;
c) компьютеры, хранящие и кодирующие информацию;
d) компьютеры, пересылающие информацию по запросам.

12. Типичная структура электронного письма:

a) заголовок, тема сообщения, ФИО адресата;
b) заголовок, тема сообщения, тип письма, адрес отправителя;
c) дата отправления, адрес, обратный адрес, тема сообщения и текст;
d) тема сообщения, адресная книга, текст и заголовок.

13. Домен — это:

a) название файла в почтовом ящике;

b) почтовый ящик узловой станции;

c) код страны;

d) короткое имя адресата.

14. Что является протокольной основой Internet:

a) система IP-адресов;

b) протоколы тестирования сетевого компьютера;

c) последовательность адресов;

d) адресная книга?

15. Из чего состоит IP-адрес:

a) адреса сети;

b) последовательности адресов;

c) протоколов;

d) адреса сети и номера хоста?

16. Какой протокол поддерживает Internet:

a) SCP/IP;

b) SCP;

c) TCP/IP;

d) QCP/IP

17. Для поддержки E-mail в Internet разработан протокол:

a) STTP;

b) SMTP;

c) SCTP;

d) SSTP.

18. Архив FTP — это:

a) сервер Archie;

b) хранилище файлов;

c) база данных;

d) WEB-сайт.

19. Взаимодействие клиент—сервер при работе на WWW происходит по протоколу

a) HTTP;

b) URL;

c) Location;

d) Uniform.

a) Какие программы не являются броузерами WWW: Mosaic;

b) Microsoft Internet Explorer;

c) Microsoft Outlook Express;

d) Netscape Navigator?

Раздел 2.

20. HTML - это:
    a) язык гипертекстовой разметки;
    b) язык структурной разметки;
    c) язык редактирования;
    d) язык программирования.
21. Все ли элементы HTML-разметки заканчиваются тегом конца?
    a) существуют неполные элементы разметки, у которых нет тега конца;
    b) элементы вида </...> не имеют тега конца;
    c) в HTML все элементы имеют тег конца элемента.
22. Неверным является утверждение:
    a) регистр букв для указания значения атрибутов имеет значение;
    b) регистр букв для указания имени атрибутов имеет значение;
    c) регистр букв для указания имени тега имеет значение.
23. Найдите неверный фрагмент кода:
    a) <html> <body> Гиперссылка </body> </html>;

b) <html> <body> Гиперссылка<a> <body> <html> ;
c) </html> </body> <ahref="doc.htm">Гиперссылка </body> </html>.

24. Отметьте верные утверждения:
a) все теги HTML состоят из начального и конечного компонентов;
b) каждому тегу необходимо указывать атрибуты;
c) порядок задания атрибутов для тега не имеет значения;
d) значения атрибутов не могут содержать символы "кавычек".

25. Что означает DTD (Document Type Declaration)? <!DOCTYPE HTML PUBLIC "- //W3C//DTD HTML 4.0 Transitional//EN">
a) указание браузеру соблюдать стандарт HTML 4.0 при отображении документа;
b) указывает, что документ соответствует стандарту HTML 4.0;
c) говорит о необходимости привести документ к стандарту HTML 4.0, используя английский язык.

26. Что является элементами контейнера?
a) тег начала элемента разметки;
b) тег конца элемента разметки;
c) содержимое контейнера.

27. Содержат ли конечные теги какие-либо атрибуты?
a) конечные теги никогда не содержат атрибутов;
b) содержат всегда;
c) понятия "конечный тег" не существует.

28. Какой тэг определяет тело документа HTML?
a) HTML;
b) BODY;
c) HEAD;
d) META.

29. Какой тег служит для именования документа в World Wide Web?
a) BASE;

b) META;
c) HEAD;
d) TITLE.

30. Какой тэг предназначен для размещения описателей стилей?
a) STYLE ;
b) HEAD;
c) BODY;
d) ISINDEX.

31. Какой тэг определяет заголовок документа HTML?
a) HTML;
b) BODY;
c) HEAD;
d) ISINDEX.

32. Какой тэг служит для размещения кода JavaScript?
a) STYLE;
b) META ;
c) SCRIPT.

33. Выберите вариант корректной установки цвета фона страницы.
a) <body bgcolor="yellow">;
b) <body color="yellow">;
c) <body background="yellow">;
d) <head bgcolor="yellow">.

34. Найдите ошибочное определение гиперссылки.
a) <a HREF=in.htm TARGET=left>int ;
b) <a HREF=in.htm>in ;
c) <a TARGET=in.htm href=new>in.

35. Какой из приведенных тегов позволяет создавать нумерованные списки?
a) DT;
b) DL;
c) UL;
d) OL.

36. Какой из приведенных тегов позволяет создавать ненумерованные списки?
a) DT;
b) DL;
c) UL;

d) OL;

37. Какие из приведенных тегов позволяют создавать списки определений?
    a) DT;
    b) DL;
    c) UL;
    d) OL.
38. Какие из приведенных тегов HTML позволяют изменять параметры шрифта?
    a) FONT;
    b) H1;
    c) HEAD;
    d) BODY.
39. Какие из приведенных тегов являются тегами, управляющими формой отображения текста?
    a) <SUP>;
    b) <Q>;
    c) < I>;
    d) < EM>.
40. Какие из приведенных тегов HTML начинают вывод текста с новой строки на странице?
    a) NOBR;
    b) P;
    c) BR;
    d) H1.
41. Что можно задавать с помощью атрибута HREF, тега <A>?
    a) ссылку на другой документ;
    b) ссылку на текущий документ;
    c) URL объекта.

42. Какой тег позволяет вставить графическое изображение в HTML-документ?
    a) INPUT;
    b) IMG;
    c) A HREF;
    d) BACKGROUND.

43. Какие основные значения атрибута ALIGN используются для определения положения изображения относительно окружающего текста?
    a) left;
    b) right;
    c) top;
    d) bottom;
    e) baseline.
44. Какие атрибуты принадлежат тегу <IMG>?
    a) ALT, ALIGN, USEMAP, HSPACE, VSPACE, BORDER;
    b) ALT, COLOR, USEMAP, HSPACE, VSPACE, BORDER;
    c) MAP, ALIGN, WIDTH, HSPACE, VSPACE, BORDER.
45. Какой атрибут тега <IMG> задает горизонтальное расстояние между вертикальной границей страницы и изображением?
    a) HSPACE;
    b) VSPACE;
    c) BORDER.
46. Укажите правильный вариант определения изображения в качестве гиперссылки.
    a) <A HREF="адрес файла"> <IMG SRC="image.gif"></A>;
    b) <A HREF="image.gif"></A>;
    c) <A HREF="адрес файла"> <IMG="image.gif"></A>.
47. Какие из приведенных фрагментов кода выводят изображение без рамки?
    a) <img src="image.jpg" width=100 height=100>;
    b) <img src="image.jpg" width=100 height=100>;
    c) <img src="image.jpg" width=100 height=100 border=0>.
48. Какой из приведённых ниже фрагментов кода выведет в окне обозревателя изображение размером 100 на 200 пикселей?
    a) <IMG SRC="image.gif" ALT="изображение" HSPACE="100" VSPACE="200">;
    b) <IMG SRC="image.gif" ALT="изображение" WIDTH="100" HEIGHT="200" HSPACE="10" VSPACE="10">;
    c) <IMG SRC="image.gif" ALT="изображение" WIDTH="100" HEIGHT="100" HSPACE="100" VSPACE="200">.

49. Выберите корректный пример вставки изображения в HTML-документ.
    a) <INPUT TYPE=image>;
    b) <IMG SRC="1.gif"> ;
    c) <BODY BACKGROUNDIMAGE="1.gif">.
50. Какой атрибут тега <IMG> указывает файл изображения и путь к нему?
    a) ALT;
    b) SRC;
    c) ALIGN.
51. Как задать цвет фона для строки таблицы?
    a) с помощью атрибута BGCOLOR тега <TD>;
    b) с помощью атрибута BACKGROUND тега <TABLE>;
    c) с помощью атрибута BACKGROUND тега <TR>;
    d) с помощью атрибута BGCOLOR тега <TR>.
52. Что определяет атрибут CELLPADDING у элемента разметки TABLE?
    a) расстояние между ячейками;
    b) расстояние от содержимого до границы ячейки;
    c) ширину ячейки;
    d) ширину границы.
53. Что определяет атрибут BACKGROUND у элемента разметки TABLE?
    a) расстояние между ячейками;
    b) расстояние от содержания до границы ячейки;
    c) ширину ячейки;
    d) фоновое изображение;
    e) вид границы.
54. Какой из приведенных фрагментов кода выводит таблицу с двумя строками и двумя ячейками в каждой строке?
    a) <table><tr><td></td><td></td></tr><tr> <td></td>
       <td></td></tr></table>;
    b) <table><tr> <td colspan=3> </td></tr></table>;
    c) <table><tr> <td></td><td></td> <td></td> </tr></table>
    d) <table><tr>;
       <td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>.
55. С помощью какого атрибута указать в таблице ширину промежутков между ячейками в 20 пикселей? …

56. Как объединить несколько ячеек таблицы?
    a) с помощью атрибута ROWSPAN;
    b) с помощью атрибута COLSPAN;
    c) с помощью атрибута CELLPADDING;
    d) с помощью атрибута CELLSPACING.
57. Что определяет атрибут BACKGROUND у элемента разметки TABLE? …
58. С помощью каких тегов описывается таблица?
    a) <TABLE>;
    b) < TR>;
    c) < OL>;
    d) < TD>;
    e) < BODY>.
59. В каких примерах корректно описан элемент TR?
    a) < TR><TD>ячейка1</TR></TD>;
    b) < TR><TD>ячейка1</TD></TR>;
    c) < TR><TH>ячейка1<TH>ячейка2</TR>.
60. С помощью какого атрибута элемента FORM указывается адрес, куда отправлять данные формы?
    a) HREF;
    b) LOCATION;
    c) ACTION;
    d) TARGET.
61. Какие контейнеры используются для задания элементов формы?
    a) INPUT;
    b) SELECT;
    c) TEXTAREA;
    d) SUBMIT;
    e) FORM.
62. Какие атрибуты из перечисленных принадлежат тэгу INPUT:
    a) Type;
    b) Value;
    c) Write.
63. Какие из приведенных тегов позволяют создавать элементы с множественным выбором?
    a) < SELECT>;
    b) < INPUT TYPE=RADIO>;

c) < INPUT TYPE=CHECKBOX>.

64. С помощью какого контейнера задается форма?...

65. Какие теги используются для создания текстовых полей ввода в форме?

a) < INPUT TYPE=text>;
b) < TEXTAREA>;
c) < INPUT TYPE=password>;
d) < SELECT>;
e) < OPTION>.

66. Какой атрибут элемента INPUT отвечает за вид элемента ввода формы?...

67. С помощью каких элементов можно создавать прокручивающиеся списки в формах? …

68. Какой из перечисленных элементов формы является необходимым для передачи формы на сервер?

a) < INPUT TYPE=submit NAME=a1>;
b) < TEXTAREA NAME=a1>;
c) < INPUT TYPE=text NAME=a1>;
d) < INPUT TYPE=reset>.

69. Какой из приведенных фрагментов кода создает радио-кнопки?

a) < input type=radio name=a1 value=1><input type=radio name=a1 value=2>
b) < input type=checkbox name=a1 value=1><input type=checkbox name=a1 value=2><input type=text name=a1 value=2>
c) < input type=radiobutton name=a1 value=1><input type=radiobutton name=a1 value=2>

Раздел 3

Вопрос 1. Как расшифровывается CSS?

- Common Style Sheets
- Computer Style Sheets
- Cascading Style Sheets

Вопрос 2. Выберите синтаксически корректное CSS правило.

- div:[font_size-2em]
- div [font-size:2em]
- div {font-size:2em}
- div {font_size:2em}

Вопрос 3. Укажите CSS свойство позволяющее устанавливать размер шрифта?

- font-weight
- font-size
- size
- weight

Вопрос 4. Укажите селектор позволяющий выбрать все элементы div имеющие атрибут id='wrap'

- div-wrap
- div id.wrap
- div.wrap
- div#wrap

Вопрос 5. С помощью какого тэга можно подключить к HTML документу внешний файл стилей?

- <style>
- <link>
- <meta>
- <css>

Вопрос 6. С помощью какого CSS свойства можно оформить границу элемента?

- border
- padding
- margin
- outline

Вопрос 7. Выберите свойство с правильно заданным значением цвета.

- color:#000000
- color:00-00-00
- color:%00-00-00
- color:#00:00:00

Вопрос 8. Выберите перечень содержащий единицы измерения, которые используются в CSS для установки размера шрифта.

- em, px
- kg, g, cm
- %, dm
- y, cm, px

Вопрос 9. Выберите существующее CSS свойство для оформления списков.

- list-color
- list-width
- list-style
- list-height

Вопрос 10. Выберите CSS свойство позволяющее скрыть элемент.

- display
- disappear
- hide
- show

Вопрос 11. Какие виды позиционирования элементов существуют в CSS.

- flow, none, show, shift
- absolute, relative, static, fixed
- slip, relating, attached, static

Вопрос 12. Укажите CSS свойство позволяющее изменять прозрачность элементов.

- display
- opacity
- transparency

Вопрос 13. Выберите перечень содержащий только существующие CSS свойства фона.

- background-color, background-repeat, background-attachment
- background-repeat, background-position, background-picture
- background-align, background-image, background-attachment

Вопрос 14. Выберите псевдо-класс позволяющий оформить ссылки, на которые наведен курсор мыши.

- :hover
- :link
- :visited
- :active

Вопрос 15. Какое CSS свойство позволяет управлять порядком наложения элементов?

- index
- z-index
- stack
- order

Вопрос 16. Выберите перечень содержащий только безопасные шрифты

- Arial, Times New Roman, Verdana
- Arial Black, Calibri, Georgia
- Comic Sans MS, Impact, Castellar
- Georgia, Century, Garamond

Вопрос 17. Какое CSS свойство было использовано для превращения первой таблицы во вторую?

Таблица 1	
Ячейка 1	Ячейка 2
Ячейка 3	Ячейка 4

Таблица 2

Ячейка 1	Ячейка 2
Ячейка 3	Ячейка 4

- border-width
- border-collapse
- border-visibility

Вопрос 18. Укажите CSS свойства, которые были использованы для оформления абзаца ниже.

Данный абзац был оформлен с помощью CSS.

- font-color, size и text-decoration
- letter-spacing и color
- color, font-size и text-decoration
- font-color, font-size и font-decoration

Вопрос 19. С помощью какого CSS свойства был скрыт второй абзац?

Первый абзац.

Третий абзац.

- visibility:hidden
- display:none
- hide:all

Вопрос 20. Какие CSS свойства использовались для оформления границы элемента ниже.

Элемент оформленный с помощью CSS

- border-type и border-color
- border-color, border-style, border-width
- border-color, border-weight, border-type
- border-type, border-style, border-width
- mouseclick

Приложение 3

КОНТРОЛЬНЫЕ РАБОТЫ

К-1
Вариант 1

Задание 1. Доступ к файлу **boom.gif**, находящемуся на сервере **light.com**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) com Г) / Ж) light
Б) :// Д) boom
В) .gif Е) http

Задание 2. Доступ к файлу **fish.gif**, находящемуся на сервере **cafe.com**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) http Г) .gif Ж) cafe.
Б) :// Д) fish
В) com Е) /

Задание 3. Доступ к файлу **home.jpg**, находящемуся на сервере **travels.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .jpg Г) / Ж)travels.
Б) http Д) home
В) :// Е) ru

Задание 4. Доступ к файлу **look.jpg**, находящемуся на сервере **birthday.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) look Г) http Ж) birthday.
Б) :// Д) .jpg
В) ru Е) /

Вариант 2.

Задание 1. Доступ к файлу **monkey.jpg**, находящемуся на сервере **animals.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) http Г) / Ж) .jpg
Б) :// Д) monkey
В) animals. Е) ru

Задание 2. Доступ к файлу **book.jpg**, находящемуся на сервере **biblioteka.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .jpg
Б) ://
В) biblioteka.
Г) http
Д) book
Е) /
Ж) ru

Задание 3. Доступ к файлу **teach.ru**, находящемуся на сервере **school.org**, осуществляется по протоколу **ftp**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .ru
Б) :/
В) ftp
Г) /school
Д) /
Е) .org
Ж) teach

Задание 4. Доступ к файлу **fox.htm**, находящемуся на сервере **animal.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .htm
Б) animal
В) /
Г) ://
Д) http
Е) .ru
Ж) fox

Вариант 3.

Задание 1. Доступ к файлу **rus.doc**, находящемуся на сервере **obr.org**, осуществляется по протоколу **https**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) obr.
Б) /
В) org
Г) ://
Д) doc
Е) rus.
Ж) https

Задание 2. Доступ к файлу **Mozart.mp3**, находящемуся на сервере **classic.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) ru
Б) .mp3
В) /
Г) classic.
Д) Mozart
Е) http
Ж) ://

Задание 3. Доступ к файлу **widow.mp3**, находящемуся на сервере **spiders.ru**, осуществляется по протоколу **http**. Фрагменты адреса

файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) /	Г) ://	Ж) widow
Б) http	Д) ru	
В) spiders.	Е) .mp3	

Задание 4. Доступ к файлу **flag.jpg**, находящемуся на сервере **rus.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) flag	Г) rus.	Ж) ru
Б) ://	Д) .jpg	
В) http	Е) /	

Вариант 4

Задание 1. Доступ к файлу jazz.mp3, находящемуся на сервере music.com, осуществляется по протоколу http. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) http	Г) .mp3	Ж) ://
Б) com	Д) /	
В) jazz	Е) music.	

Задание 2. Доступ к файлу **kassa.xls**, находящемуся на сервере **magazin.com**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) kassa	Г) com	Ж) .xls
Б) ://	Д) http	
В) /	Е) magazin.	

Задание 3. Доступ к файлу **books.xls**, находящемуся на сервере **biblio.com**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) /	Г) http	Ж) books
Б) biblio.	Д) com	
В) ://	Е) .xls	

Задание 4. Доступ к файлу **txt.html**, находящемуся на сервере **ru.net**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) http
Б) ://
В) .html
Г) txt
Д) /
Е) ru
Ж) .net

К-2
Вариант 1.

Задание 1. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	рассказы | повести | Толстой | Чехов
Б	рассказы | повести
В	рассказы & повести
Г	рассказы | повести | Толстой

Задание 2. В таблице приведены запросы к поисковому серверу. Расположите коды запросов в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	рассказы & Толстой & Чехов & повести
Б	(рассказы | Толстой) & (Чехов | повести)
В	(рассказы | Толстой) & Чехов
Г	Толстой | рассказы | повести

Задание 3. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос

А	повести & рассказы & Толстой
Б	(повести & рассказы) | Толстой
В	повести & рассказы
Г	Чехов & повести & рассказы & Толстой

Задание 4. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	мопсы & (паспорт | родословная)
Б	мопсы & родословная
В	(мопсы | пудели) & (паспорт | родословная)
Г	мопсы & пудели & родословная & паспорт

Задание 5. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	паспорт | родословная
Б	(мопсы | пудели) & (паспорт | родословная)
В	мопсы & пудели & паспорт & родословная
Г	мопсы | пудели | паспорт | родословная

Вариант 2.

Задание 1. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	мопсы

Б	(мопсы & паспорт) & (родословная | пудели)
В	мопсы & паспорт
Г	мопсы | пудели

Задание 2. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	мопсы & пудели & паспорт & родословная
Б	пудели & (родословная | паспорт)
В	(пудели | мопсы) & (родословная | паспорт)
Г	мопсы & пудели & (родословная | паспорт)

Задание 3. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	(гуппи | скалярии) & (кормление | лечение)
Б	гуппи | скалярии | кормление | лечение
В	гуппи & скалярии & кормление & лечение
Г	гуппи | скалярии

Задание 4. В таблице приведены запросы к поисковому серверу. Расположите обозначения запросов в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	гуппи | скалярии
Б	(гуппи & скалярии) & (кормление | лечение)

В	гуппи
Г	гуппи & скалярии

Задание 5. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Баскетбол
Б	Футбол & Баскетбол & Волейбол
В	Волейбол | Баскетбол
Г	Футбол | Баскетбол | Волейбол

Вариант 3.

Задание 1. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Машина | Автобус
Б	Машина
В	Метро| Автобус| Машина
Г	Автобус & Машина

Задание 2. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Курица | Колбаса | Яблоки
Б	(Курица | Яблоки) & Колбаса
В	Курица | Яблоки
Г	Курица & Колбаса & Яблоки

Задание 3. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Турция & Доминикана & Анапа
Б	(Турция | Анапа) & Доминикана
В	Турция | Доминикана | Анапа
Г	Турция | Анапа

Задание 4. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	(Зеленый | Красный) & Желтый
Б	Зеленый | Желтый | Красный
В	Зеленый & Желтый & Красный
Г	Красный | Зеленый

Задание 5. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Фрукты | Овощи
Б	Фрукты & Мясо & Овощи
В	(Фрукты | Овощи) & Мясо
Г	Фрукты | Мясо | Овощи

Вариант 4.

Задание 1. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Дуб & Клён & Берёза
Б	Дуб | Клён
В	Дуб & Берёза
Г	(Дуб & Берёза) | Клён

Задание 2. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Волк | Коза
Б	(Волк | Коза) & Капуста
В	Волк & Коза & Капуста
Г	Волк & Капуста

Задание 3. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г.

Расположите коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Отдых & Работа & Безделье
Б	Отдых | Безделье
В	Отдых | Работа | Безделье
Г	Отдых & Безделье

Задание 4. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&»:

Код	Запрос
А	Квартира | Пентхаус | Дом
Б	Квартира & Дом
В	Квартира | Дом
Г	Квартира & Пентхаус & Дом

Задание 5. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке **возрастания** количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» – «&».

Код	Запрос
А	(Умножение & Вычитание) | Сложение
Б	(Умножение | Вычитание) & Сложение
В	Умножение | Вычитание | Сложение
Г	Умножение & Вычитание & Сложение

Вариант 1.

Задание 1. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 248.137.249.32

Маска: 255.255.252.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
255	249	248	224	137	32	8	0

Задание 2. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 194.128.208.64

Маска: 255.255.224.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	192	194	208	224	255

Задание 3. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 128.194.208.64

Маска: 255.255.224.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	192	194	208	224	255

Задание 4. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 64.128.194.208

Маска: 255.255.224.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	192	194	208	224	255

Задание 5. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 130.131.132.128

Маска: 255.255.192.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	129	130	131	192	255

Задание 6. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 208.128.193.64

Маска: 255.255.224.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	192	193	208	224	255

Задание 7. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 64.128.194.208

Маска: 255.255.224.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	192	194	208	224	255

Задание 8. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 208.64.195.128

Маска: 255.255.224.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	192	195	208	224	255

Вариант 2.

Задание 1. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 130.132.131.64

Маска: 255.255.192.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	130	131	132	192	255

Задание 2. По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 130.131.132.64
Маска: 255.255.192.0
При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	130	131	132	192	255

Задание 3. По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 130.132.131.128
Маска: 255.255.192.0
При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	130	131	132	192	255

Задание 4. По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 129.130.131.128
Маска: 255.255.192.0
При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	129	130	131	192	255

Задание 5. По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 129.131.130.64
Маска: 255.255.192.0
При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	129	130	131	192	255

Задание 6. По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 129.131.130.64
Маска: 255.255.192.0
При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	129	130	131	192	255

Задание 7. По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 64.128.208.194
Маска: 255.255.224.0
При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	192	194	208	224	255

Задание 8. По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 229.37.229.32
Маска: 255.255.224.0
При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
255	249	229	224	37	32	8	0

Вариант 3.

Задание 1. По заданным IP-адресу узла и маске определите адрес сети.
IP-адрес узла: 128.64.208.32
Маска: 255.255.224.0
При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	32	64	128	192	208	224	255

Задание 2. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 32.128.208.64

Маска: 255.255.224.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	32	64	128	192	208	224	255

Задание 3. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 32.64.208.224

Маска: 255.255.192.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	32	64	128	192	208	224	255

Задание 4. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 32.64.208.224

Маска: 255.255.128.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	32	64	128	192	208	224	255

Задание 5. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 64.32.128.224

Маска: 255.255.192.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	32	64	128	192	208	224	255

Задание 6. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 208.64.192.32

Маска: 255.255.128.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	32	64	128	192	208	224	255

Задание 7. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 208.32.128.64

Маска: 255.255.192.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	32	64	128	192	208	224	255

Задание 8. По заданным IP-адресу сети и маске определите адрес сети.

IP-адрес узла: 224.24.254.134

Маска: 255.255.240.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
255	240	232	224	234	24	8	0

Вариант 4.

Задание 1. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 224.37.249.37

Маска: 255.255.240.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
255	249	240	224	37	32	8	0

Задание 2. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 224.23.253.138

Маска: 255.255.240.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
255	253	240	224	138	23	8	0

Задание 3. По заданным IP-адресу сети и маске определите адрес сети:

IP-адрес узла: 224.32.255.131

Маска: 255.255.240.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
255	249	224	240	137	32	8	0

Задание 4.По заданным IP-адресу сети и маске определите адрес сети:

IP-адрес узла: 107.173.203.135

Маска: 255.255.252.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	3	107	135	173	200	203	252

Задание 5. По заданным IP-адресу узла и маске определите адрес сети:

IP-адрес узла: 202.135.130.135

Маска: 255.255.248.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	120	128	130	135	202	248	255

Задание 6. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 102.9.140.219

Маска: 255.255.192.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
9	128	255	192	102	0	140	219

Задание 7. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 200.135.210.135

Маска: 255.255.248.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	102	110	130	135	200	208	210

Задание 8. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 100.135.223.130

Маска: 255.255.252.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы, без использования точек.

A	B	C	D	E	F	G	H
0	100	130	135	170	173	220	223

К-4

Вариант 1.

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места – нули. Обычно маска записывается по тем же правилам, что и IP-адрес – в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Задание 1. Для узла с IP-адресом 115.181.92.48 адрес сети равен 115.181.80.0. Чему равно значение третьего слева байта маски? Ответ запишите в виде десятичного числа.
Задание 2. Для узла с IP-адресом 195.227.196.12 адрес сети равен 195.227.192.0. Чему равно наибольшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.
Задание 3. Для узла с IP-адресом 203.155.196.98 адрес сети равен 203.155.192.0. Найдите наибольшее возможное количество единиц в двоичной записи маски подсети.
Задание 4. Для узла с IP-адресом 119.83.208.27 адрес сети равен 119.83.192.0. Каково наименьшее возможное количество единиц в разрядах маски?
Задание 5. Для узла с IP-адресом 147.192.92.64 адрес сети равен 147.192.80.0. Чему равно значение третьего слева байта маски? Ответ запишите в виде десятичного числа.
Задание 6. Для узла с IP-адресом 119.83.200.27 адрес сети равен 119.83.192.0. Каково наибольшее возможное количество единиц в разрядах маски?

Вариант 2.

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места – нули. Обычно маска записывается по тем же правилам, что и IP-адрес – в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.
Задание 1. Для узла с IP-адресом 119.83.208.27 адрес сети равен 119.83.192.0. Каково наибольшее возможное количество единиц в разрядах маски?
Задание 2. Для узла с IP-адресом 136.128.196.48 адрес сети равен 136.128.192.0. Чему равно наибольшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.
Задание 3. Для узла с IP-адресом 218.159.208.24 адрес сети равен 218.159.192.0. Чему равно наибольшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Задание 4. Для узла с IP-адресом 93.138.70.47 адрес сети равен 93.138.64.0. Каково наименьшее возможное общее количество единиц во всех четырёх байтах маски? Ответ запишите в виде десятичного числа.

Задание 5. Для узла с IP-адресом 93.138.70.47 адрес сети равен 93.138.64.0. Каково наибольшее возможное общее количество единиц во всех четырёх байтах маски? Ответ запишите в виде десятичного числа.

Задание 6. Для узла с IP-адресом 57.179.208.27 адрес сети равен 57.179.192.0. Каково наибольшее возможное количество единиц в разрядах маски?

Вариант 3.

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места – нули.

Обычно маска записывается по тем же правилам, что и IP-адрес – в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Задание 1. Для узла с IP-адресом 93.138.96.47 адрес сети равен 93.138.64.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Задание 2. Для узла с IP-адресом 93.138.88.47 адрес сети равен 93.138.80.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Задание 3. Для узла с IP-адресом 93.138.161.94 адрес сети равен 93.138.160.0. Какое **наибольшее** количество нулей может быть в двоичной записи маски?

Задание 4. Для узла с IP-адресом 93.138.161.94 адрес сети равен 93.138.160.0. Какое **наименьшее** количество нулей может быть в двоичной записи маски?

Задание 5. Для узла с IP-адресом 93.138.164.49 адрес сети равен 93.138.160.0. Для скольких различных значений маски это возможно?

Задание 6. Для узла с IP-адресом 93.138.161.49 адрес сети равен 93.138.160.0. Для скольких различных значений маски это возможно?

Вариант 4.

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули. Адрес сети получается в результате применения поразрядной конъюнкции к заданным IP-адресу узла и маске. Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Задание 1. Для узла с IP-адресом 153.82.140.123 адрес сети равен 153.82.136.0. Определите третий слева октет маски подсети. Ответ запишите в виде десятичного числа.

Задание 2. Для узла с IP-адресом 98.162.201.94 адрес сети равен 98.162.192.0. Для скольких различных значений маски это возможно?

Задание 3. Для узла с IP-адресом 98.162.198.94 адрес сети равен 98.162.192.0. Для скольких различных значений маски это возможно?

Задание 4. Для узла с IP-адресом 111.81.88.27 адрес сети равен 111.81.80.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Задание 5. Для узла с IP-адресом 98.162.71.94 адрес сети равен 98.162.64.0. Для скольких различных значений маски это возможно?

Задание 6. Для узла с IP-адресом 98.162.77.94 адрес сети равен 98.162.64.0. Для скольких различных значений маски это возможно?

Приложение 4.

ДОПОЛНИТЕЛЬНЫЕ ЗАДАЧИ

Задание 1. Доступ к файлу **moscow.jpg**, находящемуся на сервере **city.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) .ru	Г) .jpg	Ж) city
Б) http	Д) moscow	
В) /	Е) ://	

Задание 2. Доступ к файлу org.docx, находящемуся на сервере rez.ua, осуществляется по протоколу http. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) ua	Г) /	Ж) org
Б) .docx	Д) rez.	
В) ://	Е) http	

Задание 3. Доступ к файлу table.xls, находящемуся на сервере sch.com, осуществляется по протоколу https. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) com	Г) sch.	Ж) /
Б) https	Д) .xls	
В) ://	Е) table	

Задание 4. Доступ к файлу zveri.jpg, находящемуся на сервере image.ru, осуществляется по протоколу http. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) zveri	Г) http	Ж) ://
Б) ru	Д) .jpg	
В) image.	Е) /	

Задание 5. Доступ к файлу ptizi.jpg, находящемуся на сервере image.ru, осуществляется по протоколу http. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) http	В) image.	Д) ptizi	Ж) .jpg
Б) ru	Г) /	Е) ://	

Задание 6. Доступ к файлу paris.mp3, находящемуся на сервере love.com, осуществляется по протоколу http. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) ://	В) love.	Д) com	Ж) http
Б) /	Г) .mp3	Е) paris	

Задание 7. Доступ к файлу **rus.doc**, находящемуся на сервере **obr.org**, осуществляется по протоколу **https**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) obr.	В) org	Д) doc	Ж) https
Б) /	Г) ://	Е) rus.	

Задание 8. Доступ к файлу **table.xls**, находящемуся на сервере **home.ru**, осуществляется по протоколу **ftp**. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) home	В) .ru	Д) table	Ж) /
Б) ://	Г) ftp	Е) .xls	

Задание 9. Доступ к файлу **htm.txt**, находящемуся на сервере **com.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) com	В) /	Д) .ru	Ж) htm
Б) http	Г) .txt	Е) ://	

Задание 10. Доступ к файлу **hello.jpg**, находящемуся на сервере **home.info**, осуществляется по протоколу **ftp**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) hello	В) ftp	Д) home.	Ж) ://
Б) .jpg	Г) info	Е) /	

Задание 11. Доступ к файлу **com.xls**, находящемуся на сервере **tt.com**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) com	В) com.	Д) tt.	Ж) ://
Б) xls	Г) http	Е) /	

Задание 12. Доступ к файлу **happy.htm**, находящемуся на сервере **sibline.ru**, осуществляется по протоколу **http**. Фрагменты

адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) /	В) .htm	Д) .ru	Ж) http
Б) sibline	Г) ://	Е) happy	

Задание 13. Доступ к файлу **tests.rar**, находящемуся на сервере **olympiada.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) tests	В) /	Д)	Е) ://
Б) http	Г) .ru	olympiada	Ж) .rar

Задание 14. Доступ к файлу **karl.htm**, находящемуся на сервере **april.ru**, осуществляется по протоколу **http**. Фрагменты адреса файла закодированы буквами от А до Ж Запишите последовательность этих букв, кодирующую адрес указанного файла в сети интернет.

А) http	Г) april	Ж) ://
Б) karl	Д) .htm	
В) /	Е) .ru	

Задание 15. На сервере **GorodN.ru** находится почтовый ящик **wait_for_mail**. Фрагменты адреса электронной почты закодированы буквами от А до Е. Запишите последовательность букв, кодирующую этот адрес.

А) GorodN	В) for_	Д) wait_
Б) mail	Г) .ru	Е) @

Задание 16. Почтовый ящик **teacher** находится на сервере **shkola1.mos.ru**. В таблице фрагменты адреса электронной почты закодированы цифрами от 1 до 6. Запишите последовательность цифр, кодирующую этот адрес.

1) @	3) shkola1	5) teacher
2) .ru	4) .	6) mos

Задание 17. Доступ к файлу **net.txt**, находящемуся на сервере **html.ru**, осуществляется по протоколу **http**. В таблице фрагменты адреса файла

закодированы цифрами от 1 до 7. Запишите последовательность цифр, кодирующую адрес указанного файла в сети интернет.

1) .ru	3) html	5) /	7) .txt
2) ://	4) net	6) http	

Задание 18. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке **возрастания** количества страниц, которые найдёт поисковый сервер по каждому запросу. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» – «&».

Код	**Запрос**
А	(Умножение | Сложение) & Вычитание
Б	Умножение & Сложение & Вычитание
В	(Умножение & Сложение) | Вычитание
Г	Умножение | Сложение | Вычитание

Задание 19. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	алгебра & физика
Б	алгебра | физика | биология
В	алгебра & физика & биология
Г	физика | биология

Задание 20. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос

А	химия & физика
Б	история | физика
В	история & химия & физика
Г	история | физика | химия

Задание 21. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код – соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые найдёт поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	Квартира & Коттедж & Таунхаус
Б	Квартира | Коттедж | Таунхаус
В	(Квартира |Коттедж) & Таунхаус
Г	(Квартира & Коттедж) |Таунхаус

Задание 20. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	Книга | Журнал | Буклет
Б	(Книга & Журнал) |Буклет
В	Книга & Журнал & Буклет
Г	(Книга |Журнал) & Буклет

Задание 21. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код – соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке убывания количества страниц, которые найдёт поисковый сервер по каждому запросу.

Код	Запрос

А	Река | Болото | Озеро
Б	(Река & Болото) |Озеро
В	Река & Болото & Озеро
Г	(Река | Болото) & Озеро

Задание 22. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код – соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке убывания количества страниц, которые найдёт поисковый сервер по каждому запросу.

Код	Запрос
А	(Река | Болото) & Озеро
Б	Река & Болото & Озеро
В	(Река & Болото) | Озеро
Г	Река | Болото | Озеро

Задание 23. Приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Запишите в таблицу коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	(Париж & Москва) | Прага
Б	Париж & Москва & Прага
В	(Париж | Москва) & Прага
Г	Париж | Прага

Задание 24. Приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Запишите в таблицу коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	(Париж & Москва) | Прага

Б	Париж | Москва | Прага
В	(Париж |Москва) & Прага
Г	Париж & Москва & Прага

Задание 25. Приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Запишите в таблицу коды запросов слева направо в порядке убывания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	Война & Мир
Б	Война | (Мир & Толстой)
В	Война | Мир | Толстой
Г	Война & Мир & Толстой

Задание 26. Приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Запишите в таблицу коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	Литература | История | Экзамен
Б	Экзамен & Литература
В	Экзамен | История
Г	История & Литература & Экзамен

Задание 27. Приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке **возрастания** количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	Солнце | Воздух | Вода
Б	(Солнце & Воздух) | Вода

В	Солнце & Воздух
Г	Солнце & Воздух & Вода

Задание 28. Приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке **возрастания** количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	Хоккей & Волейбол
Б	Футбол & Хоккей & Волейбол
В	Футбол | Хоккей | Волейбол
Г	Футбол | (Хоккей & Волейбол)

Задание 29. Приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке **возрастания** количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	Киплинг & Маугли & Слоненок
Б	Киплинг & Слоненок
В	Киплинг | Маугли | Слоненок
Г	Киплинг | Слоненок

Задание 30. Приведены запросы к поисковому серверу. Для каждого запроса указан его код — соответствующая буква от А до Г. Расположите коды запросов слева направо в порядке **возрастания** количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	Есенин & Фет
Б	(Есенин & Фет) | Тютчев
В	Есенин & Фет & Тютчев

Г	Есенин | Фет | Тютчев

Задание 31. В таблице приведены запросы к поисковому серверу. Для каждого запроса указан его код – соответствующая буква от А до Г. Расположите коды запросов в порядке **возрастания** количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
А	Рим & Париж & Лондон
Б	Лондон | Рим
В	Рим & Лондон
Г	Рим | Париж | Лондон

Задание 32. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 16.8.192.181

Маска: 255.255.192.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	255	16	181	192	64	128	8

Задание 33. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 249.17.100.96

Маска: 255.255.224.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	17	54	96	100	224	249	255

Задание 34. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 111.32.96.96

Маска: 255.255.192.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	32	64	96	100	111	192	255

Задание 35. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 199.145.130.81

Маска: 255.255.128.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	81	113	128	130	145	199	225

Задание 36. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 199.145.130.81

Маска: 255.255.128.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	81	113	128	130	145	199	225

Задание 37. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес: 240.144.182.134

Маска: 255.255.248.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H

0	134	144	176	182	240	248	6

Задание 38. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 32.130.201.117

Маска: 255.255.240.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
240	201	192	130	117	32	9	0

Задание 39. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 135.21.171.214

Маска: 255.255.248.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
170	168	160	135	132	16	21	0

Задание 40. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 135.12.170.217

Маска: 255.255.248.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
0	12	16	132	135	160	168	170

Задание 41. Для узла с IP-адресом 111.81.88.168 адрес сети равен 111.81.88.160. Найдите наименьшее значение последнего байта маски. В ответ запишите наименьшее значение последнего байта маски в десятичном виде.

Задание 42. Для узла с IP-адресом 117.191.37.84 адрес сети равен 117.191.37.80. Чему равно наименьшее возможное значение последнего (самого правого) байта маски? Ответ запишите в виде десятичного числа.

Задание 43. Для узла с IP-адресом 98.162.71.94 адрес сети равен 98.162.71.64. Чему равно наименьшее возможное значение последнего (самого правого) байта маски? Ответ запишите в виде десятичного числа.

Задание 44. Для узла с IP-адресом 98.162.71.94 адрес сети равен 98.162.71.64. Чему равно наибольшее возможное значение последнего (самого правого) байта маски? Ответ запишите в виде десятичного числа.

Задание 45. Узлы с IP-адресами 98.162.71.150 и 98.162.71.140 находятся в одной сети. Чему равно наибольшее количество возможных единиц в маске этой сети?

Задание 46. Узлы с IP-адресами 98.162.78.100 и 98.162.78.90 находятся в одной сети. Чему равно наибольшее количество возможных единиц в маске этой сети?

Задание 47. Узлы с IP-адресами 140.37.235.224 и 140.37.235.192 находятся в одной сети. Определите последний байт маски сети. Количество возможных единиц в маске этой сети должно быть наибольшим.

Задание 48. Для узла с IP-адресом 111.81.176.27 адрес сети равен 111.81.160.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Задание 49. Узлы с IP-адресами 98.162.71.151 и 98.162.71.155 находятся в одной сети. Чему равно наибольшее количество возможных единиц в маске этой сети?

Задание 50. Узлы с IP-адресами 98.162.78.139 и 98.162.78.154 находятся в разных сетях. Чему равно наименьшее количество возможных единиц в масках этих сетей?

Задание 51. Узлы с IP-адресами 98.162.71.151 и 98.162.71.155 находятся в разных сетях. Чему равно наименьшее количество возможных единиц в масках этих сетей?

Задание 52. Для узла с IP-адресом 113.191.37.168 адрес сети равен 113.191.37.160. Чему равно наибольшее возможное количество единиц в последнем байте этой маске сети?

Задание 53. Для узла с IP-адресом 113.191.169.34 адрес сети равен 113.191.160.0. Каково наименьшее возможное количество единиц в разрядах маски?

Задание 54. Для узла с IP-адресом 15.51.208.15 адрес сети равен 15.51.192.0. Чему равно наибольшее возможное значение третьего слева байта маски?

Содержание

Введение 3

РАЗДЕЛ I. IP-АДРЕСА ИХ ПРИМЕНЕНИЕ И СТРУКТУРА..... 5

Тема 1.1. Знакомство с IP-адресами 5
Тема 1.2. Адресация в интернете и адрес файла на сервере 13
Тема 1.3. Поиск информации в интернете 20
Тема 1.4. Подсчет количества адресов в сети 36
Тема 1.5 Определение адреса и маски сети 48

Раздел II. HTML (HyperText Markup Language) 62

Тема 2.1. Структура HTML-документа 62
Тема 2.2. Работа с отступами и шрифтами 67
Тема 2.3. Работа со списками 71
Тема 2.4. Работа со ссылками 74
Тема 2.5. Графика и мультимедиа 79
Тема 2.6. Формы HTML и ввод данных 82

Раздел III. Каскадные таблицы стилей 96

Тема 3.1. Предназначение стилей. Синтаксис CSS и способы их подключения 96
Тема 3.2. Параметры CSS для фона. Параметры текста в CSS. 102
Тема 3.3. Шрифты в CSS 109

Тема 3.4. Границы в CSS. Отступы в CSS. Поля в CSS. 113
Тема 3.5. Размеры элементов в CSS..... 119
Тема 3.6. Параметры форматирования в CSS..... 122
Тема 3.7. Позиционирование в CSS 127
Практическая работа №1. Работа с каскадными таблицами стилей. 132
Практическая работа №2. Экспорт стилей и валидация. 138

Литература..... 139

Интернет-ресурсы 139
Приложение 1. 141
Приложение 2. 147
Приложение 3 161
Приложение 4. 182

Эльканова Айшат Амыровна,
Лепшокова Аланида Нориевна

Компьютерные сети и интернет.
Учебно-методическое пособие

План университета 2019

Редактор Н.В. Ефрюкова
Корректор Л.Ю. Козлова
Компьютерная вёрстка и набор А.М. Узденова

Подписано в печать 29.10.2019
Формат 60х84/16
Бумага офисная

Объем: 11 усл.печ.л.

Тираж 100 экз.

Издательство Карачаево-Черкесского государственного университета:
369202, г. Карачаевск, ул. Ленина, 29.
ЛР №040310 от 21.10.1997.

Отпечатано в типографии Карачаево-Черкесского
государственного университета
369202, г. Карачаевск, ул. Ленина, 46.