

---

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ**

Государственное образовательное учреждение высшего образования  
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И  
РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

**ИНФОРМАЦИОННО-ПОИСКОВЫЕ ЯЗЫКИ**

*Учебно – методическое пособие по курсу «Информационно-поисковые языки» по  
выполнению практических работ и самостоятельной работы для студентов ВУЗа*

Томск  
2018

Пособие составлено в соответствии с тематикой практических работ и самостоятельной работы по дисциплине «Информационно-поисковые языки». Пособие содержит темы и содержание практических работ, методические указания к их проведению. Для преподавателей, аспирантов, студентов и магистрантов.

СОСТАВИТЕЛЬ: Е.А. Шельмина

## СОДЕРЖАНИЕ

Раздел 1. Практические работы .....	4
Практическая работа №1 .....	4
Практическая работа №2 .....	6
Практическая работа №3 .....	7
Практическая работа №4 .....	11
Практическая работа №5 .....	13
Практическая работа №6 .....	15
Практическая работа №7 .....	16
Раздел 2. Самостоятельная работа .....	22
Список литературы .....	22

## Раздел 1. Практические работы

### Практическая работа №1 Информационно-поисковые языки

**Цель работы:** получить навыки работы с различными браузерами.

#### Информационный поиск документов в системе Интернет

Интернет – это глобальная сеть компьютерных сетей, соединенных каналами связи и передающих информацию друг другу по определенным правилам, называемых протоколами. В качестве каналов связи могут использоваться и обычные телефонные линии. Для подключения компьютера к сети через телефонную линию используется устройство, называемое модемом (от слов модуляция/демодуляция – преобразование цифровой информации, обрабатываемой компьютером, в аналоговые телефонные сигналы). Предоставляют доступ к сети Интернет организации, называемые провайдерами (от английского provide - обеспечивать).

В настоящее время наиболее интенсивно развивающимся информационным ресурсом и службой Интернета является World Wide Web (WWW) или «Всемирная паутина». В основе системы WWW лежат четыре основных понятия: гипертекст; единый формат документов (HTML); единая система адресации (URL); программы для просмотра документов (браузеры);

«Всемирная паутина» сегодня это не только огромное количество разнообразной информации, но и новые мультимедийные технологии ее передачи, позволяющие говорить об Интернете как о «виртуальной» реальности - новом способе общения и сосуществования людей.

#### Навигация в Интернете

После того, как установлена связь компьютера пользователя с сетью, например, через сеансовое подключение по телефонному каналу к серверу провайдера, для путешествия по Интернету нужно запустить специальную программу, называемую браузером (от англ. browse – листать, просматривать) или обозревателем. Браузеры позволяют просматривать гипертекстовые документы, получаемые из Интернета.

Гипертекст – это текст со ссылками, перемещение по которым позволяет обращаться к разнообразной мультимедийной информации: текстам, изображениям, звуковым и видео-файлам, расположенным где-либо в Интернете.

Гиперссылки в окне браузера обычно выделяются подчеркиванием или цветом. Основным признаком возможности перехода по ссылке является преобразование указателя мыши над ней в изображение руки человека. Если щелкнуть по ссылке кнопкой мыши, то начинается прием и воспроизведение документа, адрес которого указан в ссылке. В новом документе также могут присутствовать ссылки. Перемещаясь по ним можно путешествовать по Интернету, щелкая мышкой на нужных ссылках. Гипертекстовое содержимое WWW называется Web-страницами. Они содержат текст, графику и ссылки на другие документы и разнообразные мультимедийные файлы, расположенные в Интернете. Тематически сгруппированные Web-страницы, расположенные на WWW-серверах, называются сайтами. Web-страницы создаются с помощью языка разметки гипертекстовых документов – HTML (HyperText Markup Language) и обычно имеют расширение htm. Инструкции языка HTML (их называют тегами) одинаково интерпретируются различными браузерами, что обеспечивает одинаковое представление Web-страниц на совершенно различных компьютерах в Сети. Web-страницы можно создавать и в простом текстовом редакторе, и в специализированных редакторах, оформляя обычный текст с использованием тегов и включая с их помощью гиперссылки. Полезную информацию при работе с браузером пользователь может получить из строки состояния, которая находится в нижней части окна.

Работа с браузером обязательно включает в себя умение разбираться в надписях, появляющихся в этой строке. В процессе перемещения по Интернету в строку состояния выводятся сообщения об адресах источников информации, режиме ожидания, готовности запрошенного документа и ряд других полезных сведений. Самый очевидный способ получить документ из Интернета заключается в указании адреса нужного документа в поле ввода «Адрес».

Необходимые адреса либо берутся из справочников, либо конструируются пользователем, исходя из понимания структуры адреса и интуиции, либо находятся с помощью специальных средств поиска информации в Интернете – информационно-поисковых систем.

### **Структура адресов WWW**

Каждый из множества компьютеров, работающих в Интернете, имеет свой собственный уникальный адрес – так называемый IP-адрес. Это цифровой адрес, состоящий из четырех групп цифр, например, 194.226.214.104 – адрес Федерального портала «Российское образование». Для удобства IP-адресам поставлены в соответствие символьные или доменные адреса. Домен – это множество компьютеров, имеющих общую часть имени. Составные части доменного адреса образуют иерархическую систему, и самый последний (правый) называется именем домена верхнего уровня. Среди доменов верхнего уровня различают географические и тематические. Географические доменные имена верхнего уровня – двухбуквенные. Они определяют принадлежность владельца имени к сети конкретной страны, например, ru (Россия) в доменном адресе сайта «Российское образование». Тематические адреса дают возможность представить сферу деятельности их владельцев: com – коммерческие фирмы, например, www.microsoft.com; edu – образовательные учреждения.

Информационным ресурсам ставят в соответствие URL-адреса (от англ. Uniform Resource Locator – универсальный адрес ресурса). URL-адрес состоит из двух частей, разделяемых двоеточием. Первая (левая) часть указывает на то, к какому типу принадлежит ресурс и как получить к нему доступ, т.е. определяет конкретный сетевой протокол. Вторая часть URL-адреса сообщает, где расположен искомый ресурс. Таким образом, URL содержит информацию не только о том, где данный ресурс расположен, но и как к нему следует обращаться. Как правило, без особой необходимости нет надобности указывать префикс http://. Браузер сам определит, по какому протоколу следует связываться с сервером, доменное имя которого введено в адресной строке, и добавит необходимые символы, причем для доступа к ресурсам WWW будет выбран именно протокол http (hyper text transfer protocol) – оптимизированный для работы с гипертекстовыми ресурсами Интернета. Иногда явное указание протокола, по которому осуществляется доступ к информационному ресурсу, является необходимым, поскольку возможны варианты доступа к ресурсу в зависимости от указанного в URL протокола.

### **Проблема кодировок кириллицы**

Информация в компьютере хранится и обрабатывается в цифровом виде. Исторически сложилось, что имеются различные таблицы кодировок, т.е. правила сопоставления символам соответствующих им цифровых кодов, для русского и других национальных алфавитов: Windows 1251, KOI8R и др. Вследствие этого порой текст в окне браузера оказывается невозможно прочесть. Обычно затруднение можно преодолеть, выбрав подходящую кодировку через меню «Вид» / «Кодировка». Если это не помогает, нужно попытаться найти гиперссылку – часто они имеются на стартовой странице сайта – позволяющую получить документ в «читаемой» кодировке, если, разумеется, разработчик этого сайта такую возможность предусмотрел.

### **Работа с браузерами**

**Браузер** – программа, служащая для просмотра Web – документов, т. е. обеспечивающая переход на другой объект в соответствии с гиперссылкой.

Web-документы хранятся в файлах. Создаются эти файлы с помощью одних программ, а просматриваются с помощью программ просмотра (браузеров). Существует много программ-браузеров. У них одна основная функция – обеспечить отображение Web-документов.

Наиболее известные браузеры, которые обладают самыми широкими возможностями при отображении Web-страниц – Internet Explorer, Google Chrome, Mozilla Firefox, Opera, Яндекс Браузер.

Каждый из вышеперечисленных браузеров имеет свой собственный внешний вид и навигацию, но цель у них одна и та же: правильно и быстро загружать веб-страницы. Как и большинство современных программ, браузеры используют графический пользовательский интерфейс (GUI), что означает вы можете перемещаться и выполнять действия с помощью мыши.

### **Задания**

1. Изучите структуру нескольких браузеров.
2. Сделайте скриншоты интерфейса каждого браузера. Укажите стрелками и подпишите: адресную строку, поисковую строку, панель задач, кнопки навигации, полосы прокрутки.
3. Опишите как: сделать стартовой (домашней) страницей <http://search.skydns.ru/> в каждом браузере; сделать закладку на понравившийся сайт или страницу; удалить историю посещений (журнал посещений).

## **Практическая работа №2**

### **Принципы функционирования информационно-поисковых систем**

**Цель работы:** освоить методы поиска информации в сети Интернет, получить представление о возможностях поиска с использованием языка запросов.

Информационно-поисковые системы – системы, позволяющие хранить, обрабатывать различные каталоги, списки, справочники, т.е. представляют собой электронный вариант соответствующих бумажных аналогов.

В Интернет обращаются за определенной информацией. Чтобы открыть нужную Web-страницу, надо иметь либо ее адрес, либо другую страницу со ссылкой на нее. Если нет ни того ни другого, обращаются к поисковым системам. Поисковая система представляет собой специализированный Web-узел. Поисковые системы классифицируются по методам поиска.

Специальные программы-роботы круглосуточно сканируют пространство Интернет на предмет обнаружения новых документов. Новые документы просматриваются, выдаются ключевые слова и краткое содержание. Выявленная информация размещается в БД поисковой системы, упорядочивается (индексируется) и классифицируется по темам. При поступлении запроса на поиск от пользователя поисковая система находит ответ в своей БД и выдает пользователю ранжированный по релевантности список адресов серверов с аннотациями, на которых “по мнению поисковой системы” находится искомая информация. Из вышесказанного следует, что индексные поисковые системы – это системы, в которых все операции производятся автоматизировано (например, [www.yandex.ru](http://www.yandex.ru), [www.rambler.ru](http://www.rambler.ru)).

Поисковый индекс обеспечивает поиск по заданным ключевым словам. В результате поиска формируется набор гиперссылок на Web-странице, содержащие указанные термины. Поисковые индексы предоставляют грандиозную широту поиска.

Различают следующие виды информационно-поисковых систем: юридические, отраслевые, электронные справочники, электронные обучающие системы.

К средствам поисковых систем относится язык запросов, он различен для разных систем, но имеет определенные характерные особенности. Рассмотрим язык запросов одной из популярных систем Яндекс. Характерной особенностью этой поисковой машины является

учет морфологии языка, т. е. поиск осуществляется по всем формам слова или слов в запросе. Например: если задано слово “лить”, то в результате поиска будут предложены и документы со словами “льет”. Стоит отметить, что не все поисковики обладают такой “способностью”.

### Задания

#### 1. Создание запросов:

- Составить запрос, где встречаются одновременно два слова “производительность компьютера”.
- Составить запрос, исключающий из поиска словоформы слова “Программирование”.
- Составить запрос на поиск страниц, где слово “графика” выделено в качестве ключевого понятия.

### Практическая работа №3 Создание Web-страниц

**Цель работы:** знакомство с основными тегами языка HTML, связанными с заголовками различного уровня, форматированием текста, заданием цвета.

#### Язык HTML для разработки web - странице

HyperText Markup Language (HTML) - язык разметки гипертекста - предназначен для написания гипертекстовых документов, публикуемых в World Wide Web.

Гипертекстовый документ - это текстовый файл, имеющий специальные метки, называемые тегами, которые впоследствии опознаются браузером и используются им для отображения содержимого файла на экране компьютера. С помощью этих меток можно выделять заголовки документа, изменять цвет, размер и начертание букв, вставлять графические изображения и таблицы. Но основным преимуществом гипертекста перед обычным текстом является возможность добавления к содержимому документа гиперссылок - специальных конструкций языка HTML, которые позволяют щелчком мыши перейти к просмотру другого документа.

Существует два способа создания гипертекстовых документов. Можно воспользоваться одним из WYSIWYG HTML-редакторов, для работы с которыми не требуется специальных знаний о внутренней структуре создаваемого документа. Этот способ позволяет создавать документы для WWW без знания языка HTML. HTML-редакторы автоматизируют создание гипертекстовых документов, избавляют от рутинной работы. Однако их возможности ограничены, они сильно увеличивают размер получаемого файла и не всегда полученный с их помощью результат соответствует ожиданиям разработчика. Но, безусловно, этот способ незаменим для новичков в деле подготовки гипертекстовых документов.

Альтернативой служит создание и разметка документа при помощи обычного редактора текста. При этом способе в текст вручную вставляются команды языка HTML. Создавая документы таким способом, вы точно знаете, что делаете.

Как уже отмечалось, HTML-документ содержит символьную информацию. Одна ее часть - собственно текст, т. е. данные, составляющие содержимое документа. Другая - теги, называемые также флагами разметки, - специальные конструкции языка HTML, используемые для разметки документа и управляющие его отображением. Именно теги языка HTML определяют, в каком виде будет представлен текст, какие его компоненты будут исполнять роль гипертекстовых ссылок, какие графические или мультимедийные объекты должны быть включены в документ. Графическая и звуковая информация, включаемая в HTML-документ, хранится в отдельных файлах. Программы просмотра HTML-документов (браузеры) интерпретируют флаги разметки и располагают текст и

графику на экране соответствующим образом. Для файлов, содержащих HTML-документы, приняты расширения .htm или .html.

Прописные и строчные буквы при записи тегов не различаются. В большинстве случаев теги используются парами. Пара состоит из открывающего и закрывающего тегов. Синтаксис открывающего тега: <имя\_тега [атрибуты]>

Прямые скобки, используемые в описании синтаксиса, означают, что данный элемент может отсутствовать. Имя закрывающего тега отличается от имени открывающего лишь тем, что перед ним ставится наклонная черта: </имя\_тега>.

Атрибуты тега записываются в следующем формате: имя[="значение"]. Кавычки при задании значения аргумента не обязательны и могут быть опущены. Для некоторых атрибутов значение может не указываться. У закрывающего тега атрибутов не бывает.

Действие любого парного тега начинается с того места, где встретился открывающий тег и заканчивается при встрече соответствующего закрывающего тега. Часто пару, состоящую из открывающего и закрывающего тегов, называют контейнером, а часть текста, окаймленную открывающим и закрывающим тегом, - **элементом**.

### **Структура HTML-документа**

Самым главным из тегов HTML является одноименный тег - <HTML>. Он должен всегда открывать ваш документ, так же, как тег </HTML> должен непременно стоять в последней его строке. Эти теги обозначают, что находящиеся между ними строки представляют единый гипертекстовый документ. Без этих тегов браузер или другая программа просмотра не в состоянии идентифицировать формат документа и правильно его интерпретировать.

Закрывающий тег так же важен, как и открывающий. Если, например, документ включен в электронное письмо, тег </HTML> дает команду программе просмотра прекратить интерпретацию текста, как HTML-кода.

HTML-документ состоит из двух частей: заголовка (head) и тела (body), расположенных в следующем порядке:

```
<HTML>
  <HEAD>
    ...
  </HEAD>

  <BODY>
    ...
  </BODY>
</HTML>
```

В HTML-документ можно включать комментарии, позволяющие скрыть часть текста от браузера. Все, что заключено между последовательностями символов <!-- и -->, при просмотре страницы остается невидимым. Комментарии не могут быть вложенными друг в друга.

### **Заголовок документа**

Включение в документ заголовочной части не является обязательным. Задачей заголовка является представление необходимой информации для браузера и сервера HTTP. Информация, размещенная внутри заголовка документа, обычно не выводится на экран (кроме названия документа).

Заголовок документа открывается тегом <HEAD>, который обычно следует сразу же за тегом <HTML>. Закрывающий тег </HEAD> показывает конец этого раздела, между ними располагаются остальные теги заголовка документа.

Чаще всего в заголовок документа включают парный тег <TITLE> ... </TITLE>, определяющий название документа. Многие программы просмотра используют его как заголовок окна, в котором выводят документ. Программы, индексирующие документы в сети



Интернет, используют название для идентификации страницы. Хорошее название должно быть достаточно длинным для того, чтобы можно было корректно указать соответствующую страницу, и в то же время оно должно помещаться в заголовке окна. Название документа вписывается между открывающим и закрывающим тегами.

### Тело документа

В отличие от заголовка, тело документа является обязательным элементом, так как в нем располагается весь материал вашего документа (есть только одно исключение, с которым мы познакомимся далее, - когда документ содержит вместо тела группу фреймов). Тело документа размещается между тегами <BODY> и </BODY>. Все, что размещено между этими тегами, интерпретируется браузером в соответствии с правилами языка HTML, позволяющими корректно отображать страницу на экране монитора.

Тег <BODY> не только обозначает начало содержимого документа, но и задает его основные свойства: цвет фона, текста и многое другое. Эти свойства определяются с помощью атрибутов, которые приведены ниже:

Атрибут	Назначение
ALINK	определяет цвет активной ссылки
BACKGROUND	указывает на URL-адрес изображения, которое используется в качестве фонового
BGCOLOR	определяет цвет фона документа
LINK	определяет цвет непосещенной ссылки
TEXT	определяет цвет текста
VLINK	определяет цвет посещенной ссылки

### Цветовое оформление документа

Определение цвета составных частей документа - один из первых шагов в его создании. Если этого не сделать, то будут использоваться цвета по умолчанию, определяемые установками браузера. Не существует каких-либо правил создания хорошо сбалансированной цветовой палитры документа. Нужно лишь заботиться о том, чтобы можно было прочитать текст, не испытывая при этом неудобств. При выборе цветовой палитры старайтесь поддерживать высокую контрастность текста и фона и избегайте соседства областей с близкими цветами.

Цвет может быть задан названием или шестнадцатеричным числом, определяющим цвет в модели RGB. Эта цветовая модель базируется на определении цвета как композиции трех основных оттенков цвета: красного (Red), зеленого (Green) и синего (Blue). Каждая компонента задается двузначным шестнадцатеричным числом (т. е. изменяется от 00 до FF). Затем эти значения объединяются в одно число, перед которым ставится символ # (большинство современных браузеров может распознать цвет и без указания символа #).

Следует также отметить, что в записи шестнадцатеричного значения цвета можно использовать как большие, так и маленькие латинские буквы, например, запись #00FF00 равнозначна записи #00ff00.

Ниже представлена таблица 16 стандартных цветов вместе с их шестнадцатеричными кодами.

Цвет	Код	Цвет	Код
black (черный)	#000000	silver (серебряный)	#C0C0C0
maroon (темно-бордовый)	#800000	red (красный)	#FF0000
green (зеленый)	#008000	lime (известь)	#00FF00
olive (оливковый)	#808000	yellow (желтый)	#FFFF00
navy (темно-синий)	#000080	blue (синий)	#0000FF

purple (фиолетовый)	#800080	fuchsia (фуксия)	#FF00FF
teal (сине-зеленый)	#008080	aqua (аква)	#00FFFF
gray (серый)	#808080	white (белый)	#FFFFFF

### Разделение текста на абзацы

Язык HTML предполагает, что автор документа ничего не знает о компьютере своего читателя. Читатель вправе установить любой размер окна и пользоваться любым из имеющихся у него шрифтов. Это означает, что место переноса в строке определяется только программой просмотра и установками конечного пользователя. Символы перевода строки оригинального документа игнорируются, в результате чего текст, отлично смотревшийся в окне вашего редактора, может превратиться в сплошной неудобочитаемый текст в окне программы просмотра.

Избежать этой неприятности позволяет разделение на абзацы при помощи тега <P>. Разместите его в начало каждого абзаца, и программа просмотра отделит абзацы друг от друга пустой строкой. Использование закрывающего тега </P> необязательно. Несколько стоящих подряд тегов <P> не дают дополнительного пространства между абзацами.

Тег абзаца имеет один атрибут, поддерживаемый большинством браузеров. Это атрибут ALIGN, задающий выравнивание текста в абзаце. Если этот атрибут не задан, то текст выравнивается по левому краю окна браузера. В таблице представлены возможные значения этого атрибута:

Значение	Функция
LEFT	Выравнивание текста по левой границе окна браузера
CENTER	Выравнивание по центру окна браузера
RIGHT	Выравнивание по правой границе окна браузера
JUSTIFY	Выравнивание текста по ширине окна браузера

### Форматирование гипертекста

Язык HTML поддерживает как логический (logical), так и физический (physical) стили форматирования содержимого документа. Использование логического форматирования указывает на назначение данного фрагмента текста, а при физическом форматировании досконально задается его внешний вид. По возможности стоит использовать логические стили, так как они позволяют браузеру выбрать наиболее подходящий документу вид. Использование логических стилей также поможет читателю разобраться в структуре документа. Физический стиль используется в основном программами, конвертирующими текстовые файлы, содержащие физическое форматирование, в HTML, так как логическое форматирование документа невозможно выполнить автоматически.

### Логическое форматирование

Хотя язык HTML включает теги для достижения различных шрифтовых эффектов (полужирный шрифт, курсив, подчеркнутый шрифт), не все браузеры их поддерживают. Однако все браузеры поддерживают тот или иной способ выделения текста. Поэтому использование логического форматирования текста в любом случае приведет к выделению программой просмотра различных частей текста и выявит структуру документа.

Говоря о логической разметке текста, можно выделить две основные части:

- выделение заголовков в документе;
- логическое выделение элементов текста.

Название документа, задаваемое с помощью тега <TITLE>, не выводится на экран как часть документа. Чтобы отобразить название используется один из тегов заголовка. Заголовки в типичном документе разделяются по уровням. Язык HTML позволяет задать шесть уровней заголовков: H1 (заголовок первого уровня), H2, H3, H4, H5 и H6. Заголовок

первого уровня имеет обычно больший размер и насыщенность по сравнению с заголовком второго уровня.

### Физическое форматирование

Одним из отличий HTML-документа от документа, подготовленного на печатной машинке, является возможность форматирования текста. Язык HTML позволяет автору документа выбрать понравившийся ему шрифт, подходящий размер букв, их цвет и начертание. За все эти параметры отображения текста отвечают теги физического форматирования. Они действуют на все символы, стоящие между открывающим и закрывающим тегами.

Теги	Результат
<B> </B>	полужирный
<I> </I>	курсив
<U> </U>	подчеркнутый
<S> </S>	зачеркнутый
<BIG> </BIG>	большой
<SMALL> </SMALL>	маленький
<SUP> </SUP>	верхний индекс
<SUB> </SUB>	нижний индекс

Элементы физического форматирования могут быть вложенными друг в друга, хотя конечный результат зависит от браузера. При этом нужно внимательно следить, чтобы один контейнер находился целиком в другом контейнере.

Кроме вышеперечисленных тегов в документе может использоваться тег <FONT>, позволяющий непосредственно задать размер и цвет шрифта. Элемент <FONT> представляет собой контейнер, т. е. требует как открывающего, так и закрывающего тегов, и сам может использоваться внутри любого другого текстового контейнера.

После стартового тега обязательно указание атрибутов, без которых элемент не оказывает никакого влияния на текст, помещенный в контейнер.

Атрибут FACE позволяет указать тип шрифта, которым программа просмотра выведет ваш текст (если таковым располагает пользователь). Если нужного шрифта нет, программа проигнорирует запрос и будет использовать шрифт, установленный по умолчанию.

Этот атрибут позволяет указать как один, так и несколько шрифтов (через запятую). Весь список будет просмотрен слева направо и первый из имеющихся на машине пользователя будет использован для вывода документа.

Атрибут SIZE служит для указания размера шрифта в условных единицах от 1 до 7. Считается, что размер "нормального" шрифта соответствует числу 3. Размер может быть как абсолютной величиной ( SIZE=5 ), так и относительной ( SIZE=+2 ). Во втором примере текущий размер шрифта увеличивается на 2.

Атрибут COLOR устанавливает цвет шрифта, который может быть задан как в формате RGB, так и указанием имени.

### Задания

**Задание 1.** Создать страницу о себе. Продемонстрировать все возможности форматирования текста.

### Практическая работа №4 Списки на Web-страницах

**Цель работы:** знакомство с тегами языка HTML для создания списков на web - странице.

## Списки

В настоящее время стандарты HTML поддерживают теги для списков трех различных видов: нумерованных (упорядоченных), маркированных (неупорядоченных) и списков определений. Списки и элементы списков являются блочными элементами. Это означает, что перед ними и после них автоматически добавляются пустые строки.

Язык HTML допускает вложенность любых видов списков. Для этого размещают одну пару тегов (стартовый и завершающий) внутри другой. Следует помнить о том, что все имеющиеся списки должны завершаться закрывающим тегом.

### Нумерованные списки

Нумерованные (упорядоченные) списки используют, когда важен порядок вывода элементов списка. Браузер автоматически вставляет номера элементов по порядку, в исходном HTML-тексте номера не печатаются. Если количество элементов списка изменится (в результате удаления или добавления новых элементов), то нумерация автоматически обновится.

Весь нумерованный список заключается между парой тегов <OL> и </OL>, а каждый элемент списка расположен между тегами <LI> и </LI> (закрывающий тег </LI> может отсутствовать). Тег <OL> может иметь атрибуты TYPE и START:

<OL START=n TYPE=вид\_счетчика>

Атрибут TYPE задает вид счетчика, возможные значения которого приведены в таблице, а START - начальное значение.

Значение	Функция
A	большие латинские буквы (A,B,C...)
a	маленькие латинские буквы (a,b,c...)
I	большие римские цифры (I,II,III...)
i	маленькие римские цифры (i,ii,iii...)
1	арабские цифры (1,2,3...); используется по умолчанию

### Маркированные списки

Маркированный (неупорядоченный) список используется для представления коллекции элементов, порядок вывода которых не важен. При выводе маркированных списков браузер автоматически вставляет специальные значки (маркеры), отмечающие каждый элемент списка.

Маркированный список начинается стартовым тегом <UL> и завершается тегом </UL>. Каждый элемент списка начинается с тега <LI> и завершается (необязательным) тегом </LI>. Тег <UL> имеет атрибут TYPE, определяющий внешний вид маркера: <UL TYPE=тип\_маркера>.

Значение атрибута TYPE может быть одним из следующих: disc (круг - форма по умолчанию), circle (окружность) или square (квадрат).

Атрибут TYPE применяется и в теге <LI> для изменения формы маркера перед конкретным элементом списка.

Тег <LI> обеспечивает вывод маркера и разделение элементов списка. Если хочется использовать нестандартные маркеры, то тег <LI> не указывается. Для выделения элементов списка в этом случае используются какие-либо картинки или символы, а тег <BR> обеспечивает переход к следующему элементу списка.

### Список определений

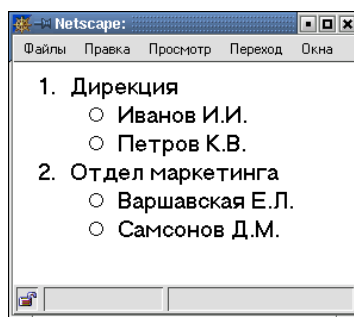
Список описаний (список определений) начинается с тега <DL> и завершается тегом </DL>. Данный список служит для создания списков типа "термин" - "описание". Термин автоматически размещается у левой границы страницы, а их определения смещены

относительно них вправо. Каждый термин обозначается тегом <DT>, а его описание - тегом <DD>.

### Вложенные списки

Любой список может быть частью другого списка, вложен в другой список. Считается полезным использование сдвигов при подготовке текста исходного HTML-документа, чтобы четко представлять уровни вложенности списков.

```
<OL>
  <LI>Дирекция</LI>
  <UL>
    <LI>Иванов И.И.</LI>
    <LI>Петров К.В.</LI>
  </UL>
  <LI>Отдел маркетинга</LI>
  <UL>
    <LI>Варшавская Е.Л.</LI>
    <LI>Самсонов Д.М.</LI>
  </UL>
</OL>
```



При выводе вложенных маркированных списков браузер автоматически проставляет маркеры перед элементами, находящимися на разных уровнях вложенности.

Для вложенных нумерованных списков браузер, к сожалению, не изменяет тип нумерации. По умолчанию каждый уровень внутри такого списка будет отмечен арабскими цифрами. Для получения списка в другом формате потребуется вручную проставить каждую метку, используя атрибут TYPE.

### Задания

**Задание 1.** Создать web-страницу на языке HTML, содержащую списки.

## Практическая работа №5

### Гипертекстовые ссылки и иллюстрации на Web-страницах

**Цель работы:** научиться использовать теги HTML для создания гиперссылок на веб-странице.

### Использование графики в HTML-документах

Рисунки и анимация могут сделать HTML-документ более привлекательным и интересным. Они не только украшают страницу, но и помогают лучше передать содержание документа. Для правильного использования графики в HTML-документе необходимо учитывать следующие факторы: многие браузеры поддерживают только графические форматы GIF и JPEG, файлы, содержащие графику, передаются медленно, некоторые пользователи не имеют графических браузеров или намерено отключают загрузку изображений.

Тег <IMG> вставляет изображение в документ, как если бы оно было просто одним большим символом. Синтаксис тега: <IMG атрибут1=значение1 ... [атрибутN=значениеN]>.

Атрибуты тега <IMG> и их значения приведены в таблице.

Атрибут	Назначение
SRC="файл"	задает URL-адрес изображения (можно указывать как абсолютный, так и относительный URL-адрес; если файл с изображением находится в той же директории, что и HTML-документ, то достаточно просто указать имя файла); этот атрибут является обязательным
ALT="текст"	задает альтернативный текст для браузеров, не поддерживающих работу с изображениями
ALIGN="тип"	задает расположение картинки относительно текста, тип может принимать следующие значения: TOP, MIDDLE, BOTTON, LEFT, RIGHT
BORDER=n	устанавливает толщину обрамления вокруг изображения в пикселах
HEIGHT=n(%)	устанавливает высоту изображения в пикселах или в процентах от высоты окна браузера
WIDTH=n(%)	устанавливает ширину изображения в пикселах или в процентах
HSPACE=n	задает свободное пространство слева и справа от изображения (в пикселах)
VSPACE=n	задается свободное пространство над и под изображением (в пикселах)

Обратите внимание, что ширина и высота изображения могут быть заданы не только в пикселах, но и в процентах от размеров окна браузера. Многие компоненты, включаемые в состав Web-страниц (изображения, таблицы, апплеты и т. д.), позволяют задавать размер в относительных единицах (т. е. в процентах). Это позволяет уменьшить зависимость внешнего вида документа от текущих установок конкретного браузера и особенностей операционной системы. Рекомендуется задавать только один из атрибутов пары "ширина-высота" изображения, иначе рисунок может быть непропорционально деформирован и изменит свой вид.

Язык HTML позволяет задать расположение изображения относительно окружающего его текста. Атрибут ALIGN может принимать следующие значения.

Значение	Функция
TOP	выравнивает одну строку по верху изображения, остальные помещает после рисунка
MIDDLE	выравнивает одну строку по середине изображения, остальные помещает после рисунка
BOTTOM	выравнивает одну строку по низу изображения, остальные помещает после рисунка
LEFT	прижимает обтекаемое текстом изображение к левой стороне окна браузера
RIGHT	прижимает обтекаемое текстом изображение к правой стороне окна браузера

### Гиперссылки

Несмотря на то, что в состав HTML-документа входят самые различные компоненты, можно сказать, что гипертекстовые ссылки - основа WWW. Если бы Web-страницы не ссылались друг на друга, содержимое Web превратилось бы в обычный набор файлов, не связанных между собой.

Для создания гипертекстовой ссылки используется пара тегов <A>... </A>. Фрагмент текста, изображение или любой другой объект, расположенный между этими тегами, отображается в окне браузера как гипертекстовая ссылка. Активация такого объекта приводит к загрузке в окно браузера нового документа или к отображению другой части текущей Web-страницы. Гипертекстовая ссылка формируется с помощью выражения: <A HREF="URL\_ресурса"> фрагмент документа </A>, HREF здесь является обязательным атрибутом, значение которого и есть URL-адрес запрашиваемого ресурса.

Текстовые указатели, т. е. фрагменты текста, являющиеся ссылками, не отличаются разнообразием внешнего вида. Обычно такой указатель представляет собой слово или слова, подчеркнутые прямой линией. Цвет указателя может регулироваться автором и установками программы просмотра.

### Задания

**Задание 1.** В текстовом файле создать HTML-страницу, которая должна содержать рисунки, ссылки, различное оформление текста (начертание, размер шрифта, выравнивание и т.д.).

### Практическая работа №6 Использование таблиц в HTML

**Цель работы:** научиться использовать теги HTML для создания таблиц на web - странице.

Теги HTML для создания таблиц первоначально предназначались для представления строк и столбцов табулированных данных. Однако дизайнеры научились с их помощью управлять разметкой веб-страниц: создавать столбцы текста, задавать интервалы между элементами и изменять внешний вид текста способами, недоступными другим тегам форматирования HTML.

Таблицы в языке HTML всегда имеют прямоугольный вид и состоят из строк, которые в свою очередь состоят из ячеек. Все языковые конструкции, описывающие компоненты создаваемой таблицы, заключаются между тегами <TABLE> и </TABLE>.

Заполнение таблицы происходит построчно. Для обозначения строки используется пара тегов <TR>... </TR>. Строка состоит из ячеек, для задания которых используют либо теги <TH>... </TH>, если эти ячейки содержат заголовки столбцов, либо теги <TD>... </TD>. Заголовки выводятся полужирным шрифтом и располагаются по центру ячейки. Данные имеют обычный шрифт и выравниваются по левой стороне ячейки. Для задания заголовка всей таблицы используются теги <CAPTION> и </CAPTION>.

#### Основные атрибуты тега <TABLE>

Назначение основных атрибутов тега <TABLE> и значения, которые они могут принимать перечислены в таблице.

Атрибут	Назначение
BORDER=n	определяет ширину рамки таблицы (в пикселах), например, BORDER=1 ; значение, равное нулю, означает отсутствие рамки
WIDTH=n	определяет ширину всей таблицы в пикселах, либо в процентах от ширины окна браузера
HEIGHT=n	определяет высоту всей таблицы в пикселах, либо в процентах от высоты окна браузера
ALIGN	задает горизонтальное выравнивание таблицы в окне браузера; имеет значения left, center и right (по умолчанию - left)
CELLPADDING=n	добавляет свободное пространство между данными внутри ячейки и ее границами; по умолчанию значение равно 2

CELLSPACING=n	добавляет свободное пространство между ячейками внутри всей таблицы (по умолчанию значение равно 2)
HSPACE=n	задает области свободного пространства указанной ширины (в пикселах) слева и справа от таблицы
VSPACE=n	задает области свободного пространства заданной высоты (в пикселах) сверху и снизу от таблицы
BGCOLOR=цвет	устанавливает цвет фона всей таблицы
BACKGROUND=URL	указывает фоновое изображение для таблицы, где URL - адрес источника (имя файла с изображением)

### Объединение ячеек

Смежные ячейки таблицы могут объединяться. Например, в таблице из нескольких столбцов все ячейки первой строки можно объединить и поместить в этой строке красивый заголовок таблицы. Возможно также объединение нескольких строк или создание пустой прямоугольной области.

Для соединения двух смежных ячеек в одном столбце нужно использовать атрибут ROWSPAN тега <TH> или <TD>, например, <TD ROWSPAN=2>.

Для объединения двух смежных ячеек в одной строке нужно использовать атрибут COLSPAN тех же тегов, например, <TD COLSPAN=2>.

### Цвет в таблицах

В HTML не предусмотрено специальных средств раскрашивания таблиц. Вы можете изменить цвет фона ячейки при помощи атрибута BGCOLOR перед размещением в ней текста или изображения, а также использовать атрибут BORDERCOLOR для изменения цвета рамки ячейки. Теги <TABLE>, <TD>, <TH> и <TR> также допускают использование в них указанных атрибутов. Таким образом, вы можете изменить цвет всей таблицы, отдельной ячейки или строки таблицы.

Значения цветов, установленные на уровне ячейки, будут перекрывать значения, установленные на уровне строки, которые в свою очередь, будут перекрывать значения, заданные на уровне всей таблицы.

### Задания

**Задание 1.** Создать веб-страницу на языке HTML, содержащую следующие таблицы:

1. 


2. 


## Практическая работа №7

### Каскадные таблицы стилей (CSS)

**Цель работы:** научиться использовать каскадные таблицы стилей при создании веб-сайтов.

### Каскадные таблицы стилей (CSS)

Каскадные таблицы стилей (Cascading Style Sheets, CSS) — это стандарт, определяющий представление данных в браузере. Если HTML предоставляет информацию о структуре документа, то таблицы стилей сообщают как он должен выглядеть.



Стиль — это совокупность правил, применяемых к элементу гипертекста и определяющих способ его отображения. Стиль включает все типы элементов дизайна: шрифт, фон, текст, цвета ссылок, поля и расположение объектов на странице.

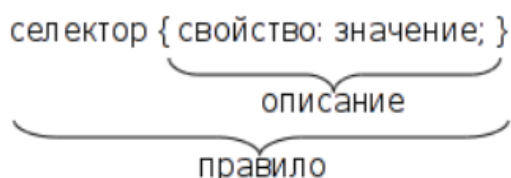
Таблица стилей — это совокупность стилей, применимых к гипертекстовому документу.

Каскадирование — это порядок применения различных стилей к веб-странице. Браузер, поддерживающий таблицы стилей, будет последовательно применять их в соответствии с приоритетом: сначала связанные, затем внедренные и, наконец, встроенные стили. Другой аспект каскадирования — наследование (inheritance), — означает, что если не указано иное, то конкретный стиль будет применен ко всем дочерним элементам гипертекстового документа.

Использование каскадных таблиц дает возможность разделить содержимое и его представление и гибко управлять отображением гипертекстовых документов путем изменения стилей.

### Общий синтаксис таблиц стилей

Таблицы стилей строятся в соответствии с определенным порядком (синтаксисом), в противном случае они не могут нормально работать. Таблицы стилей состояются из определенных частей (рис. 1):



Селектор — это элемент, к которому будут применяться назначаемые стили. Это может быть тег, класс или идентификатор объекта гипертекстового документа. Свойство определяет одну или несколько характеристик селектора. Свойства задают формат отображения селектора: отступы, шрифты, выравнивание, размеры и т.д. Значения — это фактически числовые или строковые константы, определяющие свойство селектора.

Таким образом, таблица стилей — это набор правил, задающих значения свойств селекторов, перечисленных в этой таблице. Общий синтаксис описания правила выглядит так:

```
селектор[, селектор[, ...]] {свойство: значение;}
```

Регистр символов значения не имеет, порядок перечисления селекторов в таблице и свойств в определении не регламентирован.

### Правила CSS

Итак, каскадная таблица стилей — это набор правил форматирования тегов HTML. Приведем несколько примеров написания таких правил:

Основной текст с выравниванием по ширине, абзацный отступ 30px, гарнитура (шрифт) — Serif, кегль (размер шрифта) — 14px:

```
p {
text-align: justify;
text-indent: 30px;
font-family: Serif;
font-size: 14px;
}
```

Это правило будет применено ко всем тегам <p>.

1. Синий цвет для заголовков с первого по третий уровень:

```
h1, h2, h3 {color: blue; /* тоже самое, что и #0000FF */}
```

2. Таблицы и изображения выводить без обрамления:

```
table, img {border: none;}
```

3. Ссылки в элементах списков показывать без подчеркивания:

```
li a {text-decoration: none;}
```

4. Внутренние отступы слева и справа для блоков (<div>), заголовков таблиц и ячеек таблиц установить в 10px и залить фон желтым цветом:

```
div, th, td {  
padding-left: 10px;  
padding-right: 10px;  
background-color: yellow;}
```

5. Все ссылки в документе отображать черным цветом и полужирным шрифтом, а в основном тексте и списках — обычным, а также выделять их зеленым цветом и подчеркивать только при наведении курсора (в описании правил использован псевдоэлемент a:hover).

```
a {color: black; font-weight: bold;}  
p a, li a {font-weight: normal; text-decoration: none;}  
p a:hover, li a:hover {  
color: #00FF00; text-decoration: underline;}
```

### Классы

Стандарт CSS представляет возможности создания именованных стилей — стилевых классов. Это позволяет ответить на такой, например, вопрос: Как применить разные стили к одному и тому же селектору?

Предположим, что в документе вам нужны два различных вида основного текста — один без отступа, второй — с левым отступом и шрифтом красного цвета.

Для этого нужно создать правила для каждого из них, например так:

```
p {margin-left: 0;}  
p.warn {margin-left: 40px; color: #FF00;}
```

Для применения созданного класса его имя нужно указать в атрибуте class для выбранных абзацев:

```
<p class="warn">Красный текст с отступом слева</p>
```

Общий синтаксис описания класса: селектор.имя\_класса {описание}.

При создании класса селектор можно не указывать, тогда это правило можно применять к любому селектору, поддерживающему тот же набор свойств.

### Идентификаторы

В качестве селектора может выступать идентификатор элемента гипертекста, указанный в атрибуте id. Для назначения стилей таким элементам используется синтаксис, аналогичный описанию классов, но вместо точки ставится знак # (“решетка”). Например:

```
div#content {  
position: absolute;  
top: 10px;  
left: 10%;  
right: 10%;  
border: solid 1px silver;}
```

...

```
<div id="content">Текст</div>
```

Следует помнить, что идентификаторы элементов должны быть уникальны в пределах документа.

### Группировка свойств

Группировка (grouping) состоит в объединении значений родственных свойств. При этом таблица стилей становится более компактной, но предъявляются более жесткие требования к описанию правил. Ниже приведен пример обычного стиля, задающего отступы:

```
div {margin-left: 10px;  
margin-top: 5px;
```

```
margin-right: 40px;
margin-bottom: 15px;}
```

Это же правило можно переписать с группировкой в следующем виде:

```
div {margin: 5px 40px 15px 10px;} /*порядок: top right bottom left*/
```

Оба стиля будут отображаться одинаково.

Группировка может применяться для таких свойств, как padding, font, border, background и еще некоторых (см. документацию CSS)

### Использование в веб-страницах

Существует три способа применения таблицы стилей к документу HTML:

*Встраивание (Inline)*. Этот метод позволяет применить стиль к заданному тегу HTML.

*Внедрение (Embedded)*. Внедрение позволяет управлять стилями страницы целиком.

*Связывание (Linked или External)*. Связанная таблица стилей позволяет вынести описание стилей во внешний файл, ссылаясь на который можно контролировать отображение всех страниц сайта.

### Встроенные стили

Встраивание стилей предоставляет максимальный контроль над всеми элементами веб-страницы. Встроенный стиль применяется к любому тегу HTML с помощью атрибута style следующим образом:

```
<p style="font: 12pt Courier">Это текст с кеглем 12 точек и гарнитурой Courier</P>
```

Пример:

```
<div style="font-family: Garamond; font-size: 18 pt;>
```

Весь текст в этом разделе имеет размер 18 точек и шрифт Garamond.

```
<span style="color:#ff3300;>
```

А этот фрагмент еще и выделен красным цветом.</span>

```
</div>
```

Встроенные стили полезны, когда необходима тонкая настройка отображения некоторого элемента страницы или небольшой веб-страницы.

### Внедренные стили

Внедренные стили используют тег <style>, который обычно размещают в заголовке HTML-документа (<head>...</head>):

```
<html>
```

```
<head>
```

```
...
```

```
<style>
```

```
правила CSS
```

```
</style>
```

```
...
```

```
</head>
```

```
<body>
```

```
...
```

### Связанные таблицы стилей

*Связанные (linked)*, или *внешние (external)* таблицы стилей — наиболее удобное решение, когда речь идет об оформлении целого сайта. Описание правил помещается в отдельный файл (обычно, но не обязательно, с расширением .css). С помощью тега <link> выполняется связывание этой таблицы стилей с каждой страницей, где ее необходимо применить, например так: <link rel=stylesheet href="sample.css" type="text/css">

Любая страница, содержащая такую связь, будет оформлена в соответствии со стилями, указанными в файле sample.css. Следует отметить, что файл со стилями физически может находиться на другом веб-сервере, тогда в href нужно указать абсолютный путь к нему.

### CSS-свойства: размеры, цвета, шрифты, текст

**Размеры.** Размеры в CSS можно задавать в различных единицах измерения: **pt** – пункты (типографская единица измерения шрифта), **px** – пиксель, **%** – процент. Гораздо реже используется указание размеров в миллиметрах (mm), сантиметрах (cm) и дюймах (in).

Единица измерения записывается сразу за значением без пробела: TABLE {font-size: 12pt}.

**Цвета.** В CSS цвет задается как и в HTML – 6 шестнадцатеричными цифрами или стандартными названиями цветов на английском.

**URL.** URL задаются конструкцией url(...). Например, следующий CSS-код добавляет фоновое изображение для страницы: BODY {background-image: url(images/bg.jpg);}

**Шрифты.** Шрифт – набор начертаний букв и знаков. В компьютере шрифт представляет собой файл, в котором описано, как должны отображаться на мониторе или принтере различные символы: буквы, цифры, знаки пунктуации и др. Типы шрифтов:

serif – шрифты с засечками (антиквенные), например: Times New Roman, Georgia.

sans-serif – рубленые шрифты (шрифты без засечек или гротески), типичные представители – Arial, Impact, Tahoma, Verdana;

cursive – курсивные шрифты: Comic Sans MS;

fantasy – декоративные шрифты, например: Curlz MT.

monospace – моноширинные шрифты, ширина каждого символа одинакова.

**Текст.** CSS позволяет управлять свойствами шрифта и текста.

**font-family** – задает начертание шрифта. Можно указать несколько значений через запятую. Браузер проверит первый шрифт из списка: если шрифт установлен на компьютере пользователя, то браузер применит его, если нет – перейдет ко второму шрифту и т.д. Последним в списке обычно указывается общий тип шрифта serif, sans-serif, cursive, fantasy или monospace.

**font-size** – размер шрифта. Может задаваться абсолютным значением в пунктах (pt) или пикселях (px) или в процентах (%) или в em.

**font-style** – задает начертание текста: normal (обычное), italic (курсивное) или oblique (наклонное). Курсивное начертание является специальной измененной версией шрифта, имитирующей рукописный текст с наклоном вправо. Наклонное начертание получается из обычного наклоном букв.

**font-weight** – позволяет изменить уровень жирности текста: normal (обычная), bold (полужирная). Действие аналогично тегу <b>.

**color** – задает цвет текста.

**line-height** – межстрочный интервал (интерлиньяж), указывает расстояние между строками текста. Может задаваться числом как множитель от текущего размера шрифта, в процентах, а также в пунктах (pt), пикселях (px) и других единицах измерения CSS.

**text-decoration** – задает оформление текста. Варианты: line-through (перечеркнутый), overline (линия над текстом), underline (подчеркивание), none (отключение эффектов).

**text-align** – выравнивание текста в блоке: left (по левому краю), center (по центру), right (по правому краю) или justify (по ширине).

**text-indent** – отступ первой строки. Длина отступа может задаваться в процентах (%) от ширины текстового блока, пикселях (px), пунктах (pt) и др.

**Ширина полей** и заполнения задается следующими CSS свойствами:

margin-top, margin-right, margin-bottom, margin-left – для верхней, правой, нижней, левой стороны поля.

Если для margin указать два значения через пробел, то первое из них будет задавать ширину верхнего и нижнего поля, а второе – левого и правого. Если указать три значения, то первое будет присваиваться верхнему полю, второе – левому и правое, а третье – нижнему. Наконец, при указании четырех значений, они поочередно будут указывать верхнее, правое, нижнее и левое поля.

**padding-top, padding-right, padding-bottom, padding-left** – устанавливают ширину заполнения1 сверху, справа, снизу и слева от содержимого соответственно.

**padding** – устанавливает значение сразу для всех сторон. Padding может принимать не только одно, но и 2, 3 или 4 значения.

Для margin и padding можно задавать значение auto. В этом случае браузер сам автоматически рассчитает величину полей и заполнения.

**Для границ можно задать толщину, цвет и стиль:**

**border-width** – толщина границы;

**border-color** – цвет границы (по умолчанию – черный);

**border-style** – стиль границы. Может принимать значения solid (по умолчанию), dotted, dashed, double, groove, ridge, inset или outset.

### **CSS-свойства: фон, оформление таблиц**

**Фон.** Как и в языке HTML, в CSS фоном служит заливка цветом или изображение. Фоновое изображение может быть повторяющимся.

**background-color** – устанавливает цвет фона. Пример: TD.head {background-color: #ffff00}

**background-image** – устанавливает в качестве фона изображение: BODY {background-image: url(images/bg.jpg)}

**background-attachment** – задает поведение фонового изображения при прокрутке. По умолчанию задается значение scroll – фон прокручивается вместе с содержимым. Значение fixed делает фон неподвижным.

**background-position** – начальное положение фонового изображения по горизонтали (left, center, right) и вертикали (top, center, bottom). Вместо ключевых слов можно указывать расстояние в пикселях или процентах.

**background-repeat** – указывает, в каком направлении должно размножиться фоновое изображение:

repeat – по горизонтали и вертикали (по умолчанию);

repeat-x – только по горизонтали;

repeat-y – только по вертикали;

no-repeat – отключить повторение.

### **Теги DIV и SPAN**

До сих пор мы применяли стили CSS к тегам, уже имеющим заранее заданную функцию: таблицам, заголовкам, параграфам и т.д. Но иногда нужно применить стили к фрагменту содержимого, не включенного в отдельный тег. Например, выделить фоном несколько слов в тексте.

Теги <div>...</div> и <span>...</span> используются там, где не подходит никакой другой тег. Сами по себе они не определяют никакого форматирования, но удобны для привязки к ним стилей. При этом DIV является блочным элементом, а SPAN – строчным.

Основное различие между блочными и строчными элементами заключается в следующем: строчные элементы идут друг за другом в строке текста, а блочные – располагаются один по другим. К строчным элементам относятся такие теги, как <a>, <img>, <input>, <select>, <span>, <sub>, <sup> и др. К блочным: <div>, <form>, <h1>...<h6>, <ol>, <p>, <table>, <ul> и некоторые другие.

Блочные элементы располагаются друг под другом, многие занимают всю возможную ширину. Блочные элементы могут включать в себя строчные и другие блочные. Но строчные элементы не могут содержать блочные.

Еще одним отличием является то, что для строчных элементов не работают такие свойства, как margin-top, margin-bottom, padding-top и paddingbottom.

Исключением являются теги <img>, <input>, <textarea> и <select> – для них можно задавать отступы padding-top и paddingbottom.

## **Задания**

**Задание 1.** Модифицировать разработанные в рамках практической работы №3 страницы, внедрив в них поддержку CSS. Использовать три способа подключения каскадных таблиц стилей: связанные, внедренные и встроенные.

С помощью CSS задайте следующие параметры:

- размер, цвет, шрифт текста на странице, межстрочный интервал, красная строка, выравнивание;
- фон страницы;
- при оформлении страницы использовать теги DIV и SPAN.

## **Раздел 2. Самостоятельная работа**

2.1. Проработка лекционного материала.

2.2. Подготовка к практическим занятиям согласно разделу 1.

## **Список литературы**

1. Кручинин В. В. Разработка сетевых приложений: Учебное пособие [Электронный ресурс] / В. В. Кручинин — Томск: ТУСУР, 2013. — 120 с. — Режим доступа: <https://edu.tusur.ru/publications/2835> , дата обращения: 13.05.2018

2. Ехлаков Ю. П. Основы гипертекстового представления интернет-контента: учебное пособие [Электронный ресурс] / Ю. П. Ехлаков, Э. К. Ахтямов — Томск: ТУСУР, 2017. — 181 с. — Режим доступа: <https://edu.tusur.ru/publications/7086> , дата обращения: 13.05.2018