ЛАБОРАТОРНАЯ РАБОТА №1
СТАТИЧНЫЕ САЙТЫ

Цель работы: получить навык разработки статичных сайтов с использованием HTML5 и CSS3

Задачи:

1. Установить и настроить веб-сервер.
2. Разработать и разместить на веб-сервере статичный сайт, содержащий таблицу стилей.

Результатами работы являются:

1. Разработанный статичный сайт, содержащий HTML и CSS файлы
2. Подготовленный отчет.

ПРОТОКОЛ HTTP

HTTP — это протокол клиент-серверного взаимодействия, позволяющий получать клиенту различные ресурсы, например, HTML-документы от HTTP-сервера (веб-сервера). Протокол HTTP лежит в основе обмена данными в Интернете. Полученный итоговый документ будет (может) состоять из различных поддокументов, являющихся частью итогового документа: например, из отдельно полученного текста, описания структуры документа, изображений, видео-файлов, скриптов и многого другого.

HTTP-запрос, отправлемый клиентом, состоит из глагола, обозначающего одно из действий, которое может выполнить сервер (GET, POST, OPTIONS, PUT, HEAD, DELETE, PATCH, TRACE, CONNECT), адрес запрашиваемого ресурса на сервере, версию протокола и набор возможных дополнительных заголовков, а также для ряда запросов тело запроса (полезная информация, например, значения заполненных форм). Пример HTTP-запроса:

```
GET / HTTP/1.1
Host: kf.bmstu.ru
Accept-Language: en
```

Ответ, формируемый сервером, содержит версию протокола, код ответа, фразу-состояние ответа, возможные дополнительные заголовки, для некоторых запросов возможна полезная нагрузка (какая-то передаваемая информация, например, HTML-разметка). Пример HTTP-ответа, формируемого сервером и передаваемого клиенту:

```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2021 14:28:02 GMT
Server: Apache
Accept-Ranges: bytes
Content-Length: 29812
Content-Type: text/html

<!DOCTYPE html.. (содержимое, общим размером 29812 байт)
```

В настоящий момент наиболее распространены следующие 3 веб-сервера:

Apache

Веб-сервер Apache стоял у истоков развития мирового интернета, он до сих пор лидирует в мировом рейтинге популярности. За свою долгую жизнь (а Apache ведет свою историю с 1995 года) свободный веб-сервер оброс массой модулей и «научился» разворачиваться на всевозможных платформах. До 2005 года Apache широко использовался как единый сервер для всех задач — он выполнял роли и веб-сервера, и обратного прокси, и балансировщиком нагрузки. Впрочем, сейчас его позиции пошатнулись — по мере увеличения трафика, количества подключений и объемов данных на страницах Apache перестал справляться с такой многозадачностью. Но это не значит, что Apache уже вышел из игры. Главное его преимущество — огромное количество подключаемых модулей. В сети можно найти библиотеки для любых задач, поэтому «Апач» с большой вероятностью подойдет для разработки необычного сайта. Кроме того, ненужные модули всегда можно отключить, чтобы повысить быстродействие.

Nginx

Nginx — это веб-сервер с открытым исходным кодом. Если необходимо что-то в нем подправить, всегда можно скачать исходный код и подогнать его под себя. Но в большинстве случаев это не требуется — у Nginx и без того широкий функционал, способный удовлетворить потребности не только простеньких проектов, но и сложных сайтов с огромной посещаемостью. Nginx относится к легковесным серверам. При его разработке старались учесть все недостатки более старого Apache. Код сервера подразумевает более эффективное масштабирование — с увеличением потока подключений скорость работы почти не падает. Каждый рабочий процесс Nginx способен обрабатывать по тысячам HTTP-подключений сразу.

IIS

IIS (Internet Information Services) — это веб-сервер, разработанный компанией Microsoft для своих операционных систем. Продукт полностью проприетарный и идет в комплекте с Windows. Первая

версия появилась в Windows NT и продолжает развиваться. IIS уступает Apache и Nginx, в виду архитектурной особенности и строгой работы на ОС Windows, но с другой стороны, как проприетарное решение, имеет поддержку производителя.

HTML

HTML (HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык разметки документов. Большинство веб-страниц содержат описание разметки на языке HTML. Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Пример HTML-документа:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Тестовая страница</title>
    </head>
    <body>
        <p>Это — моя страница</p>
    </body>
</html>
```

В первой строке традиционно приводится объяление типа документа. Далее в теге <html> приводится сам HTML-документ, состоящий из двух составных частей – блока <head> и блока <body>.

В блоке <head> содержится метаинформация о странице, т.е. необходимая для работы браузеру, но не присутствующая на самой рисуемой странице: информация о дополнительных зависимостях, иконки и заглавие страницы, сведения об используемой кодировке и другая метаинформация.

Блок <body> содержит непосредственно «тело» страницы, т.е. саму гиперразметку текста.

Большинство структурированных текстов состоят из параграфов и заголовков, независимо от того, читаете ли вы рассказ, или газету, или учебник, журнал и т.д.

В HTML каждый абзац заключен в элемент <p>, подобно:

```
<p>Я параграф, да, это я.</p>
```

Каждый заголовок заключен в элемент заголовка <h1>:

```
<h1>Я заголовок истории.</h1>
```

Имеется шесть элементов заголовка: <h1>, <h2>, <h3>, <h4>, <h5> и <h6>. Каждый элемент представляет разный уровень контента в документе; <h1> представляет главный заголовок, <h2> представляет подзаголовки, <h3> представляет под-подзаголовки и так далее.

Для курсивного текста в HTML используется элемент (выделение). Кроме того, чтобы сделать документ более интересным для чтения, они распознаются программами, считывающими с экрана, и произносятся другим тоном.

```
<p>Я <em>рад</em>, что ты не <em>опоздал</em>.</p>
```

Для полужирного текста в HTML используется элемент (важное значение). Помимо того, что документ становится более полезным, они распознаются программами, считывающими с экрана, и говорят другим тоном.

```
<p>Эта жидкость
    <strong>очень токсична</strong>
.</p>
<p>Я рассчитываю на тебя.
    <strong>Не </strong>
опаздывай!</p>
```

Также в HTML есть возможность создавать списки двух видов: нумерованные и ненумерованные (для каждого нового элемента будет использоваться стандартный для браузера разделитель):

Неупорядоченные:

```
<ul>
    <li>молоко</li>
    <li>яйца</li>
    <li>хлеб</li>
    <li>хумус</li>
</ul>
```

Упорядоченные

```
<ol>
    <li>Доедьте до конца дороги</li>
    <li>Поверните направо</li>
    <li>Езжайте прямо</li>
    <li>Поверните налево на третьем перекрестке</li>
    <li>Школа справа от вас, в 300 метрах</li>
</ol>
```

Для добавления в текст изображения необходимо использовать тег <img>, в аттрибутах которого необходимо указать источник изображения и замещающий текст, отображаемый в случае недоступности изображения:

```
<img src="путь к файлу" alt="альтернативный текст">
```

Для представления сложной структуры текста можно использовать таблицы. Таблица состоит из строк и столбцов ячеек, которые могут содержать текст и рисунки. Для добавления таблицы на веб-страницу используется тег <table>. Этот элемент служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются соответственно с помощью тегов <tr> и <td>. Таблица должна содержать хотя бы одну ячейку.

Допускается вместо тега <td> использовать тег <th>. Текст в ячейке, оформленной с помощью тега <th>, отображается браузером шрифтом жирного начертания и выравнивается по центру ячейки. В остальном, разницы между ячейками, созданными через теги <td> и <th> нет.

Для объединения двух и более ячеек в одну используются атрибуты colspan и rowspan тега <td>. Атрибут colspan устанавливает число ячеек объединяемых по горизонтали. Аналогично работает и атрибут rowspan, с тем лишь отличием, что объединяет ячейки по вертикали. Перед добавлением атрибутов проверьте число ячеек в каждой строке, чтобы не возникло ошибок. Пример:

```
<table border="1">
    <tr>
          <td colspan="2">Строка 1</th>
    </tr>
    <tr>
          <td>Столбец 1</td>
          <td>Столбец 2</td>
    </tr>
</table>
```

Гиперссылки

Гиперссылки позволяют нам связывать документы с любым другим документом (или ресурсом), с которым необходимо. URL-адрес может указывать на файлы HTML, текстовые файлы, изображения, текстовые документы, видео и аудиофайлы и все остальное. Если веб-браузер не знает, как отображать или обрабатывать файл, он спросит, хотите ли вы открыть файл (в этом случае обязанность открытия или обработки файла передается в соответствующее локальное приложение на устройстве) или загрузить файл.

Простая ссылка создается путем обертывания текста (или другого содержимого), который необходимо превратить в ссылку, в элемент <a>, и придания этому элементу атрибута href (который также известен

как гипертекстовая ссылка, или цель), который будет содержать веб-адрес, на который нужно указать ссылку:

```
<p>Ссылка на <a href="http://kf.bmstu.ru" title="Наш родной университет">сайт университета</a>.</p>
```

Можно ссылаться на определенную часть документа HTML (известную как фрагмент документа), а не только на верхнюю часть документа. Для этого вам сначала нужно назначить атрибут id элементу, с которым вы хотите связаться. Обычно имеет смысл ссылаться на определенный заголовок, поэтому это выглядит примерно так:

```
<h2 id="mail">Почтовый адрес</h2>
```

Затем, чтобы связаться с этим конкретным id, необходимо включить его в конец URL-адреса, которому предшествует знак решетки («якорь»), например:

```
<p>Хотите написать мне письмо? Используйте наш
    <a href="contacts.html#mail">почтовый адрес</a>
.</p>
```

При наличии атрибута download браузер не переходит по ссылке, а предложит скачать документ, указанный в адресе ссылки.

```
<a download>Ссылка</a>
```

По умолчанию, при переходе по ссылке документ открывается в текущем окне или фрейме. При необходимости, это условие может быть изменено атрибутом target тега <a>

В качестве значения используется имя окна или фрейма, заданное атрибутом name. Если установлено несуществующее имя, то будет открыто новое окно. В качестве зарезервированных имен используются следующие:

– _blank – загружает страницу в новое окно браузера.
– _self – загружает страницу в текущее окно.
– _parent – загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как _self.
– _top – отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как _self.

Блочные и строчные элементы

Исторически HTML-элементы было принято делить на блочные и строчные. Блочные элементы занимают всю ширину своего родителя (контейнера), формально создавая «блок» (отсюда и название).

Браузеры обычно отображают блочные элементы с переводом строки до и после элемента. Блочные элементы можно представить в виде стопки коробок.

Блочные элементы: , <dl>, <dt>, <fieldset>, <figcaption>, <figure>, <footer>, <form>, <h1> – <h6>, <header>, <hgroup>, <hr>, <li>, <main>, <nav>, <ol>, <p>, <pre>, <section>, <table>, <ul>, <address>, <article>, <aside>, <blockquote>, <details>, <dialog>, <dd>.

Все остальные элементы являются строчными и могут начинаться в любом месте строки.

Нововведения HTML5

Поддержка аудио и видео, элементы для замены небезопасного flash:

– <audio>
– <video>

Рисование, элемент-холст:

<canvas>

Новые атрибуты форм, такие как placeholder, require, novalidate

Использование локального хранилища данных.

Media-запросы для загрузки адаптивных стилей.

Перечень HTML-элементов: http://htmlbook.ru/html

СТИЛИЗАЦИЯ HTML. CSS

Элементам HTML можно задавать дополнительные настройки стиля с помощью атрибута style. Пример:

```
<p style="font-family:times; color:green;">
    Этот текст написан зеленым цветом шрифтом Times.
</p>
```

Однако, это неудобно при наличии на странице большого количества элементов, к которым должны быть применены одинаковые стили – дублирование кода увеличит размер страницы в разы. Для сокращения рекомендуется использовать CSS (Cascading Style Sheets). CSS можно описывать как в блоке <head>, так и в отдельном файле. Рекомендуется использовать второй вариант для сокращения передаваемых данных благодаря кешированию CSS-файлов браузером.

Как и HTML, CSS не является языком программирования. Но это и не язык разметки - это язык таблицы стилей. Это означает, что он позволяет применять стили выборочно к элементам в документах HTML.

Например, чтобы выбрать все элементы абзаца на HTML странице и изменить текст внутри них с черного на красный, можно использовать этот CSS:

```
p {
      color: red;
}
```

Структура CSS-правила:

Селектор определяет набор элементов, к которым будет применен набор последующих стилистических свойств. Через запятую можно перечислять несколько селекторов. Перечень возможных селекторов приведен в Таблице 1.1. В фигурных скобках указывается набор стилистических правил, состоящий из разделенных точкой с запятой пар из имени свойства и задаваемого ему значения.

Таблица 1.1. Типы селекторов

Имя селектора	Что выбирает	Пример
Селектор элемента (иногда называемый селектором тега или типа)	Все HTML элемент(ы) указанного типа.	p
		Выбирает все <p>
ID селектор	Элемент на странице с указанным ID на данной HTML. Лучше всего использовать один элемент для каждого ID (и конечно один ID для каждого элемента)	#my-id
		Выбирает элемент с аттрибутом id="my-id"
Селектор класса	Элемент(ы) на странице с указанным классом (множество экземпляров класса может объявляться на странице).	.my-class
		Выбирает все элементы с аттрибутом class="my-class"
Селектор атрибута	Элемент(ы) на странице с указанным атрибутом.	img[src]
		Выбирает <img src="myimage.png"> но не <img>
Селектор псевдо-класса	Указанные элемент(ы), но только в случае определенного состояния, например, при наведении курсора.	a:hover
		Выбирает <a>, но только тогда, когда указатель мыши наведен на ссылку.

Псевдоклассы:

- :link – выберет любую ссылку, которая еще не была посещена, даже те, которые уже стилизованы
- :visited – озволяет вам выбирать ссылки, которые были посещены.
- :active – соответствует элементу в момент, когда он активируется пользователем (TAB)
- :hover – срабатывает, когда пользователь наводит на элемент мышью, но не обязательно активирует его
- :focus – применяется, когда элемент (такой как input формы) получает фокус. Обычно он активируется при клике мышью пользователем или при выборе элемента с использованием клавиши "tab" на клавиатуре.
- :first-child – находит любой элемент, являющийся первым в своем родителе.
- :last-child – любой элемент, являющийся последним в его родителе.
- :nth-child() – находит один или более элементов, основываясь на их позиции среди группы соседних элементов (2n, 2n+1 и т.д.).
- :nth-last-child() – находит элемент, имеющий an+b-1 потомков после данной позиции в дереве документа, значение для n может быть положительным или нулевым, а также имеющий родительский элемент
- :nth-of-type() – находит один или более элементов с заданным тегом, основываясь на их позиции среди группы соседних элементов.
- :first-of-type – находит первого потомка своего типа среди детей родителя.
- :last-of-type – находит последнего потомка с заданным тегом в списке детей родительского элемента.
- :empty – находит любой элемент, у которого нет потомков.

– :target – представляет уникальный элемент (целевой элемент) с подходящим id URL-фрагментом (модальное окно, вкладка и т.п.)
– :checked – находит отмеченные radio-buttons и check-buttons
– :enabled – находит любой включенный элемент. Элемент включен, если его можно активировать (например, выбрать, нажать на него или ввести текст) или поставить фокус. У элемента также есть отключенное состояние, когда его нельзя активировать или сфокусировать.
– :disabled – находит любой не включенный элемент
– :after – применяется для вставки назначенного контента после содержимого элемента. Этот псевдоэлемент работает совместно со стилевым свойством content, которое определяет содержимое для вставки.
– :before – по своему действию :before аналогичен псевдоэлементу :after, но вставляет контент до содержимого элемента.
– :first-letter – определяет стиль первого символа в тексте элемента, к которому добавляется. Это позволяет создавать в тексте буквицу и выступающий инициал.
– :first-line – определяет стиль первой строки блочного текста.

Если к одному элементу одновременно применяются противоречивые стилевые правила, то более высокий приоритет имеет правило, у которого значение специфичности селектора больше. Специфичность – это некоторая условная величина, вычисляемая следующим образом. За каждый идентификатор (в дальнейшем будем обозначать их количество через a) начисляется 100, за каждый класс и псевдокласс (b) начисляется 10, за каждый селектор тега и псевдоэлемент (c) начисляется 1. Складывая указанные значения в определенном порядке, получим значение специфичности для данного селектора.

Встроенный стиль, добавляемый к тегу через атрибут style, имеет специфичность 1000, поэтому всегда перекрывает связанные и

глобальные стили. Однако добавление !important перекрывает в том числе и встроенные стили.

Если два селектора имеют одинаковую специфичность, то применяться будет тот стиль, что указан в коде ниже.

Нововведения CSS3

- Градиенты цветов
- Скругленные элементы
- Тени
- Анимация

Перечень доступных CSS-свойств: http://htmlbook.ru/css

Flexbox

CSS Flexbox — это технология для создания сложных гибких макетов за счет правильного размещения элементов на странице.

Для создания макета необходимо поместить контент внутрь flexbox-контейнера:

```
.container {
    display: flex; /* or inline-flex */
}
```

Далее необходимо установить направление основной оси

```
.container {
    flex-direction: row | row-reverse | column | column-reverse;
}
```

Выравнивание «резиновых» компонентов:

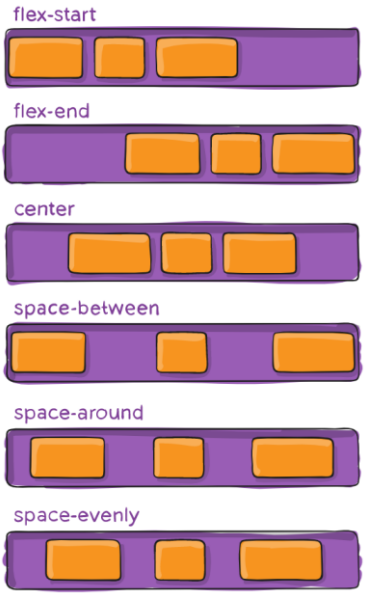


Рис. 1.1. Значение свойства justify-content для выравнивания вдоль основной оси

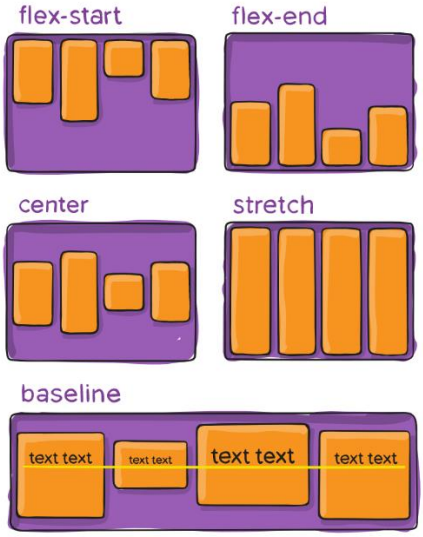


Рис. 1.2 Значения свойства align-items для выравнивания вдоль попреречной оси

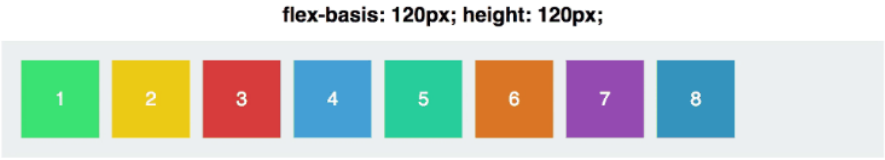
Для автоматического переноса дочерних элементов можно использовать свойство flex-wrap:

```
flex-wrap: wrap;
```

Можно сократить запись flex-direction и flex-wrap в одну строку, используя следующее свойство:

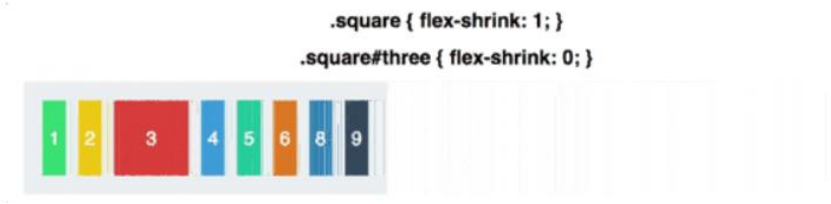
```
flex-flow: row wrap;
```

Свойство flex-basis отвечает за изначальный размер элементов до того, как они будут изменены другими свойствами CSS Flexbox:



Свойство flex-grow определяет на сколько блок может увеличиться в размерах (размеры относительные!). Значение 0 запрещает изменение размера блока.

Свойство flex-shrink определяет, насколько блоку можно уменьшиться в размере (размеры относительные!). Значение 0 запрещает изменение размера блока.



Свойство flex заменяет flex-grow, flex-shrink и flex-basis. Значения по умолчанию: 0 (grow) 1 (shrink) auto (basis):

```
.square#one {
    flex: 2 1 300px;
}
```

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Установить и сконфигурировать для локальной работы один из следующих веб-серверов на свой выбор: Apache, Nginx или IIS.

Разработать HTML-страницы статичного сайта тематики, согласно полученному варианту.

У сайта должны быть заданы заглавие и иконка сайта.

Сайт должен состоять минимум из 2-х страниц с возможностью перехода между ними.

Среди элементов обязательно наличие таблицы (с объединением некоторых ячеек), картинка, а также какой-либо из видов списков.

На сайте должна быть предусмотрена возможность скачивания произвольного PDF-документа.

Необходимо реализовать CSS-файл для стилизации сайта, обязательно должны использоваться селекторы идентификатора и класса, а также псевдокласса.

ВАРИАНТЫ ЗАДАНИЙ

1. Сайт школы
2. Сайт фитнес-центра
3. Новостной сайт
4. Метеорологический сайт
5. Сайт музыкального исполнителя
6. Сайт кинотеатра
7. Сайт отеля
8. Сайт аэропорта
9. Сайт телеканала
10. Сайт радиостанции
11. Сайт автосалона
12. Сайт ресторана
13. Сайт университета
14. Сайт библиотеки
15. Сайт театра
16. Сайт ветеренарной клиники
17. Сайт туристической фирмы
18. Сайт агентства по продаже недвижимости
19. Сайт букмекерской компании
20. Сайт биржи криптовалют

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Приведите формат HTTP-запроса и HTTP-ответа.
2. Раскройте назначение веб-сервера.
3. Опишите формат HTML-документа.
4. Опишите различие между блочным и строчными элементами.
5. Опишите способ создания гиперссылки в HTML-документе.
6. Охарактеризуйте аттрибуты id и class.
7. Приведите способы стилизации HTML-документов.
8. Опишите структуру CSS-правила.

9. Опишите и приведите примеры различным CSS-селекторам.
10. Дайте определение псевдоклассу. Приведите примеры
11. Опишите механизм приоретизации противоречивых CSS-правил.
12. Опишите механизм создания гибких макетов HTML-страниц

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 5 часов: 4 часа на выполнение работы, 1 час на подготовку и защиту отчета по лабораторной работе.

Номер варианта студенту выдается преподавателем.

Структура отчета: титульный лист, цель и задачи, формулировка задания (вариант), этапы выполнения работы (установки и настройки веб-сервера), исходный код разработанного сайта, результаты выполнения работы (скриншоты), выводы.