



Федеральное агентство по рыболовству
БГАРФ ФГБОУ ВО «КГТУ»
Калининградский морской рыбопромышленный колледж

Утверждаю
Заместитель начальника колледжа
по учебно-методической работе
А.И.Колесниченко

МДК 05.02.06 ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ

Методические указания для выполнения практических занятий
по специальности

09.02.07 Информационные системы и программирование
(специалист по информационным системам)

МО–09 02 07-ПМ.05. МДК.05.02. Тема 5.2.6. ПЗ

РАЗРАБОТЧИК	Халина Е.Н.
ЗАВЕДУЮЩИЙ ОТДЕЛЕНИЕМ	Кругленя В.Ю.
ГОД РАЗРАБОТКИ	2024
ГОД ОБНОВЛЕНИЯ	2025

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 2/52

Содержание

ВВЕДЕНИЕ	3
ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ	4
Практическое занятие №1 Применение гипертекстовых ссылок	5
Практическое занятие № 2 Построение таблиц различного вида. Обтекание таблицы текстом	8
Практическое занятие №3 Формы ввода данных.....	16
Практическое занятие № 4 Использование изображений. Карты MAP	20
Практическое занятие №5 Пример использования таблиц стилей. Теги «div» и «span»	23
Практическое занятие № 6 Шрифты в CSS.....	29
Практическое занятие №7 Поля в CSS. Списки в CSS.....	33
Практическое занятие № 8 Позиционирование в CSS	39
Практическое занятие № 9 Контекстные селекторы.....	48
Практическое занятие №10 Работа над html страницей по индивидуальному заданию	51

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 3/52

ВВЕДЕНИЕ

Методические указания для выполнения практических занятий разработаны в соответствии с рабочей программой ПМ.05 «Проектирование и разработка информационных систем» по специальности 09.02.07 «Информационные системы и программирование»

Данное методическое пособие раскрывает рекомендации по выполнению практических занятий для темы 5.2.6 Интернет программирование по теме «Введение в HTML и CSS» и предусматривает проведение 10 практических занятий-20 академических часа в соответствии с учебным планом.

Целью проведения практических занятий является формирование профессиональных компетенций согласно ФГОС, закрепление теоретических знаний и приобретение необходимых практических навыков и умений по отдельным темам курса.

Внеаудиторная самостоятельная работа выполняется в отдельных тетрадях в виде конспекта (реферата, презентации).

Перед проведением практического занятия обучающиеся должны повторить пройденный материал по данной теме, уяснить цель занятия, ознакомиться с последовательностью его проведения. Преподаватель должен проверить знания обучающихся и готовность к выполнению задания.

Для ведения записей (отчетов) по выполнению практических занятий обучающийся должен иметь отдельную тетрадь с полями. Записи должны вестись четко, аккуратно, разборчивым почерком. Каждая новая работа оформляется с новой страницы.

Для закрепления знаний по теме и подготовки к защите в конце каждой работы приводится перечень вопросов для самоконтроля. К защите обучающийся должен представить оформленный отчет по работе согласно задания с выводами по работе, должен знать теорию по данной теме, уметь пояснить, как проводилась работа и уметь анализировать полученные результаты.

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 4/52

ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

№ п/п	Наименование практических занятий	Кол-во часов
1	Применение гипертекстовых ссылок.	2
2	Построение таблиц различного вида. Обтекание таблицы текстом	2
3	Формы ввода данных	2
4	Использование изображений. Карты MAP.	2
5	Пример использования таблиц стилей. Теги «div» и «span»	2
6	Шрифты в CSS.	2
7	Поля в CSS. Списки в CSS.	2
8	Позиционирование в CSS.	2
9	Контекстные селекторы	2
10	Работа над html страницей по индивидуальному заданию	2
Итого		20

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 5/52

Практическое занятие №1 Применение гипертекстовых ссылок

Цель занятия: Применение форматирования текстовой информации.

Исходные материалы и данные:

1. Конспект лекций по теме
2. Персональный компьютер
3. Программа Notepad++

Содержание и порядок выполнения задания:

Формирование или форматирование текстовой информации представляет собой изменение вида текста, применяя различные начертания и использования эффектов. В различных средах хранения информации, формирование текста используются по-своему. В нашем случае форматирование базируются на понятии тег или дескриптор, что подразумевает, какой-то объект. Объявление тегов бывают трех видов: открывающиеся – <название тега; атрибуты> закрывающиеся – </название тега> и одиночные <название тега; атрибуты />.

Использование тегов для форматирования текста происходит так:

<название тега; атрибуты>Текст</название тега >

Мы рассмотрим самые основные теги для форматирования текстовой информации:

Код в HTML	Описание	Пример
Текст	Жирное начертание текста	Текст
<i>Текст</i>	Курсивное начертание текста	<i>Текст</i>
<sup>Текст</sup>	Верхний индекс (надстрочный)	e=mc ²
<sub>Текст</sub>	Нижний индекс (подстрочный)	H ₂ O
<pre>Текст</pre>	Текст пишется как есть, включая все пробелы	Текст
Текст	Курсивный текст	<i>Текст</i>
Текст	Жирное начертание текста	Текст
<u>Текст</u>	Нижнее подчеркивание	<u>Текст</u>

Любые теги форматирования текста можно использовать совместно друг с другом. Чтобы сделать текст одновременно жирным и курсивным шрифтом используется сочетание тегов **** и **<i>** (рис. 1.1.).

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 6/52

`<p>Он словно вырезан из каменя, стоек и неподвижен в отличие от его противников. Дух и жизненная сила в нём достигла совершенства. Но вот беда – никто не смеет принять его вызов.</p>`

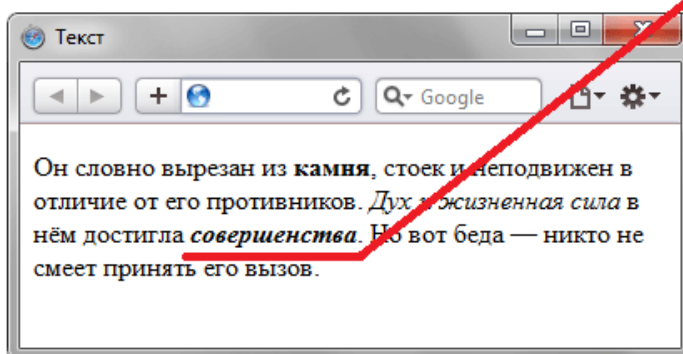


Рисунок 1.1.

Следует отметить, что теги `` и ``, также как `<i>` и `` являются не совсем эквивалентными и заменяемыми. Первый тег `` — является тегом физической разметки и устанавливает жирный текст, а тег `` — тегом логической разметки и определяет важность помеченного текста. Такое разделение тегов на логическое и физическое форматирование изначально предназначалось, чтобы сделать HTML универсальным, в том числе не зависящим от устройства вывода информации. Теоретически, если воспользоваться, например, речевым браузером, то текст, оформленный с помощью тегов `` и ``, будет отмечен по-разному. Однако получилось так, что в популярных браузерах результат использования этих тегов равнозначен.

Задание:

1. Создадим HTML-документ по шаблону. Рис.1.2.

```
<html>
  <head>
    <title></title>
  </head>

  <body>

  </body>
</html>
```

Рисунок 1.2.

2. Дадим название страницы в теге TITLE—Текстовая информация в HTML.
3. Создадим тег – заголовок 2 (`<h2></h2>`) с тем же названием, что и TITLE.
4. Создадим тег параграф – `<p></p>` и в него занесём следующий текст: HTML (от англ. HyperTextMarkupLanguage — «язык разметки гипертекста») — стандартный язык разметки документов во Всемирной паутине.

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 7/52

5. Отформатируем текст, таким образом:

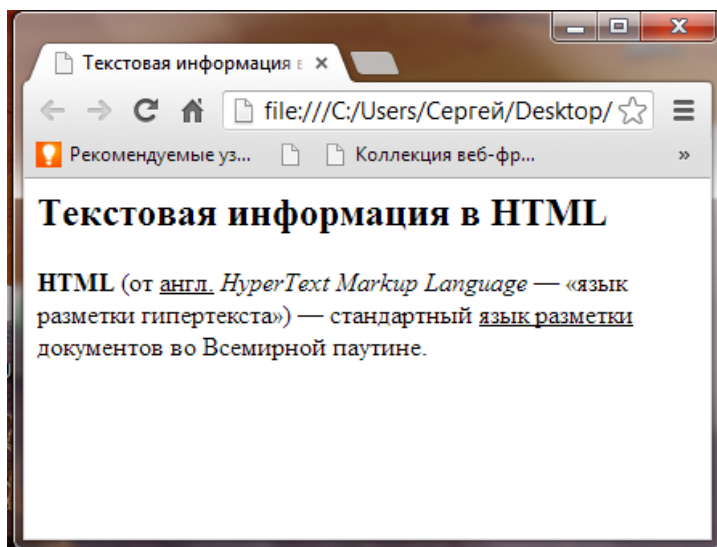


Рисунок 1.3.

Исходный код:

```
<p>
  <b>HTML</b> (от <u>англ.</u> <i>HyperText Markup Language</i> –
  «язык разметки гипертекста») – стандартный <u>язык
  разметки</u> документов во Всемирной паутине.
</p>
```

6. Добавим еще один параграф и запишем туда такой текст:

Под действием гидроксиламина альдегиды превращаются в оксимы: $\text{CH}_3\text{CHO} + \text{NH}_2\text{OH} = \text{CH}_3\text{C}(=\text{NOH})\text{H} + \text{H}_2\text{O}$

И отформатируем, используя тег ``.

7. Как же работает тег `<pre></pre>`? Давайте посмотрим добавив его и внесем в него следующий стих, самое главное записать его в код надо прям так как есть:

Улица провалилась, как нос сифилитика.

Река - сладострастье, растекшееся в слюни.

Отбросив белье до последнего листика,

сады похабно развалились в июне.

Я вышел на площадь,

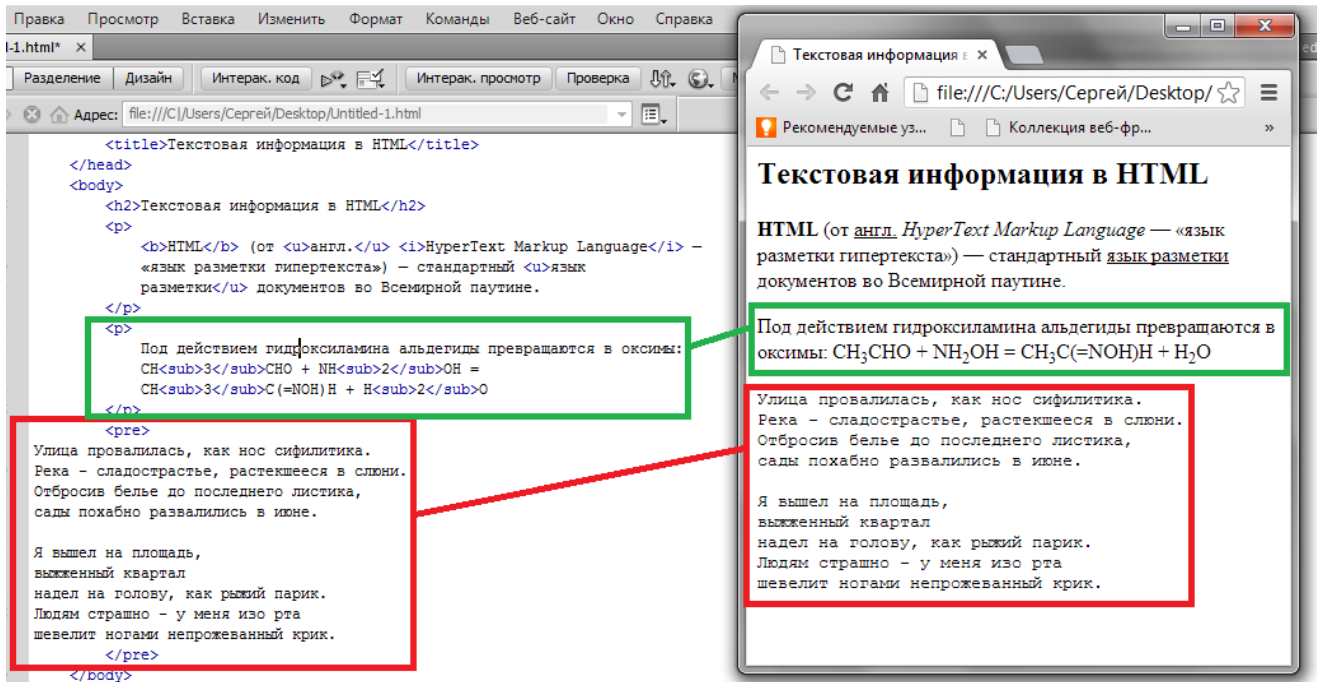
выжженный квартал

надел на голову, как рыжий парик.

Людям страшно - у меня изо рта

шевелит ногами непрожеванный крик.

Результат работы:



Содержание отчёта:

1. Наименование практического занятия.
2. Цель занятия.
3. Отчёт о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
4. Список использованных источников.
5. Выводы и предложения.

Вопросы для самопроверки:

1. Какие шаблоны существуют при создании гипертекстовых ссылок?
2. Листинг написания
3. Как сделать ссылку в пределах одного документа?

Практическое занятие № 2 Построение таблиц различного вида. Обтекание таблицы текстом

Цель занятия:

Научиться строить таблицы различного вида

Исходные материалы и данные.

Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж

1. Конспект лекций по теме
2. Персональный компьютер
3. Программа Notepad++

Содержание и порядок выполнения задания:

Существует два вида таблиц, простые и сложные. Различия сложной таблицы от простой являются в ее структуре. Сложная таблица имеет объединения, когда простая не имеет. Давайте посмотрим на примере:

Простая таблица:

Сложная таблица:

Как же происходит объединение ячеек в HTML? Объединение происходит, таким образом, мы знаем из чего, состоит начальный и одиночный тег. Он состоит из названия и атрибута. Так вот существует тег **<TD>**, который создает нам колонки 2 атрибута ROWSPAN и COLSPAN, которые имеют значения.

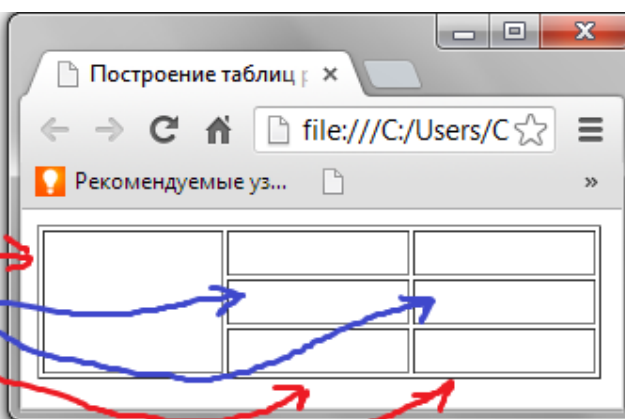
ROWSPAN – объединяет строки. При использовании надо занести в него количество объединяемых строк, указанное количество на последующей строке надо удалить.

Пример:

```

<table width="300" border="1">
<tr>
<td rowspan=3>&nbsp;&nbsp;&nbsp;</td>
<td>&nbsp;&nbsp;&nbsp;</td>
<td>&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td>&nbsp;&nbsp;&nbsp;</td>
<td>&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td>&nbsp;&nbsp;&nbsp;</td>
<td>&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>

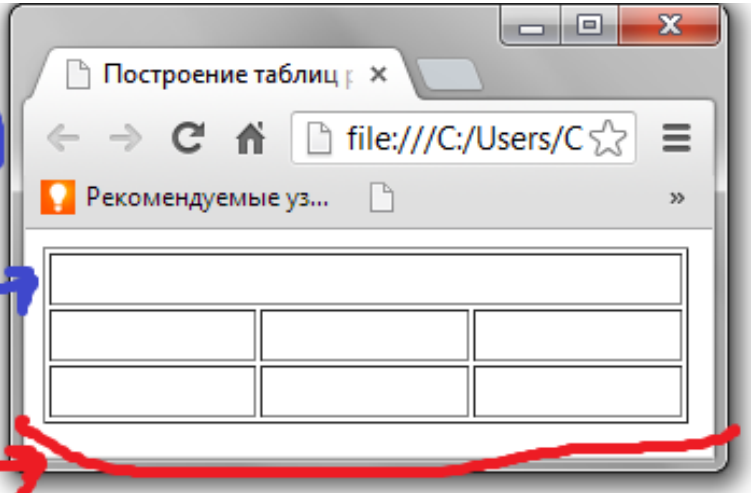
```



COLSPAN – объединяет колонки. При использовании надо занести в него количество объединяемых колонок, лишние колонки в строке надо удалить.

Пример:

```
<table width="300" border="1">
  <tr>
    <td colspan=3>&nbsp;&nbsp;&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;&nbsp;&nbsp;</td>
    <td>&nbsp;&nbsp;&nbsp;</td>
    <td>&nbsp;&nbsp;&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;&nbsp;&nbsp;</td>
    <td>&nbsp;&nbsp;&nbsp;</td>
    <td>&nbsp;&nbsp;&nbsp;</td>
  </tr>
</table>
```



Задание:

Для начала нам надо представить себе, как же будет выглядеть таблица. Сделаем набросок:

№ п/п	Наименование предмета	Распределение по предметам		Кол-во контрольных работ	Количество часов			Распределение по курсам и предметам											
		Экзаменов	Курс. проект		Всего	Из них			2 курс		3 курс		4 курс		5 курс				
						Уроки	Лаб. раб.	Курс проект	1 сем. 16 нед.	2 сем. 17 нед.	3 сем. 16 нед.	4 сем. 19 нед.	5 сем. сем.	6 сем. сем.	7 сем. сем.	8 сем. сем.			
	Факультативные предметы, консультации	Всего	Изучаемых предметов																
			Курсовых проектов																
			Экзаменов																

Следующим нашим шагом будет определение, сколько будет использоваться строк и столбцов.

№ п/п	Наименование предмета	Распределение по предметам		Количество часов				Распределение по курсам и предметам							
		Экзаменов	Курс. проект	Всего	Уроки	Лаб. раб.	Курс проект	1 сем. 16 нед.	2 сем. 17 нед.	3 сем. 16 нед.	4 сем. 19 нед.	5 сем.	6 сем.	7 сем.	8 сем.
Факкультативные предметы, консультации		Исучаемых предметов													
		Всего		Курсовых проектов											
				Экзаменов											

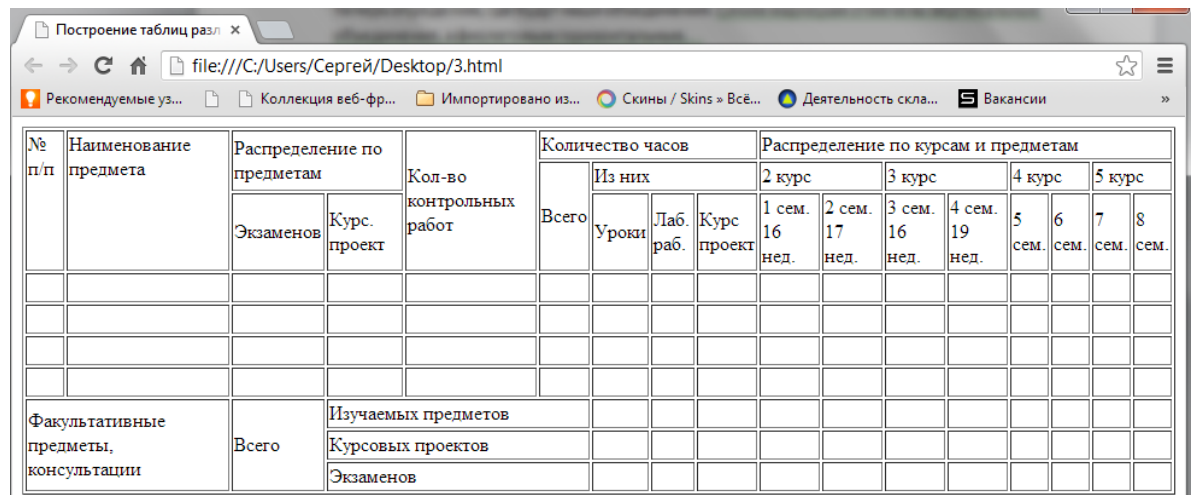
В нашем случае строк будет 10, а столбцов 17.

Теперь определим, где будут наши объединения. Синим маркерам отмечены вертикальные объединения, а фиолетовым горизонтальные.

№ п/п	Наименование предмета	Распределение по предметам		Количество часов				Распределение по курсам и предметам							
		Экзаменов	Курс. проект	Всего	Уроки	Лаб. раб.	Курс проект	1 сем. 16 нед.	2 сем. 17 нед.	3 сем. 16 нед.	4 сем. 19 нед.	5 сем.	6 сем.	7 сем.	8 сем.
Факкультативные предметы, консультации		Исучаемых предметов													
		Всего		Курсовых проектов											
				Экзаменов											

Создадим новый HTML документ и создадим таблицу с 10-ю строками и 17-ми колонками. После создания начнём редактировать таблицу.

Результат работы:



Код:

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 14/52

```

<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td ROWSPAN=3 COLSPAN=2>Факультативные предметы, консультации</td>
<td ROWSPAN=3>Всего</td>
<td COLSPAN=3>Изучаемых предметов</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td COLSPAN=3>Курсовых проектов</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td COLSPAN=3>Экзаменов</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>

```

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 15/52

```
<td>&nbsp;</td>  
</tr>  
</table>
```

Содержание отчёта:

1. Наименование практического занятия.
2. Цель занятия.
3. Отчёт о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
4. Список использованных источников.
5. Выводы и предложения.

Вопросы для самопроверки:

1. Какие теги создают столбцы таблицы?
2. Какие теги создают строки таблицы?
3. Перечислите теги для объединения ячеек?

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 16/52

Практическое занятие №3 Формы ввода данных

Цель занятия:

Научиться создавать формы и обрабатывать данные, полученные из формы.

Исходные материалы и данные.

1. Конспект лекций по теме
2. Персональный компьютер
3. Программа Notepad++

Содержание и порядок выполнения задания:

Формы предназначены в HTML для отправки данных на сервер. Объявление форм происходит тегом `<form></form>`. Саму форму вы не увидите, это не визуальный тег. Основные атрибуты формы **method** и **action**. Чем являются, эти атрибуты рассмотрим: **method** – метод отправки данных на сервер, а **action** – файл который будет принимать данные на сервере.

Для того чтобы отправить данные их надо с начало куда-то ввести. Для этого в HTML есть дополнительные теги, которые встраиваются в форму.

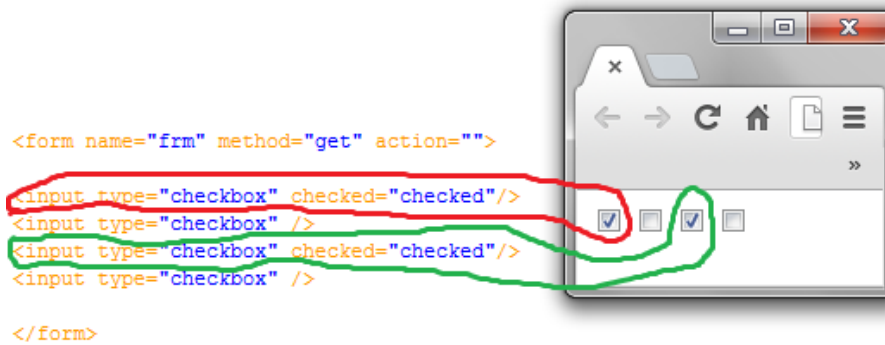
Основные теги: `<input/>`, `<button/>`, `<select></select>`, `<option></option>`.

Рассмотрим по порядку теги.

`<input/>` – имеет основные атрибуты: **type**, **value**. Данный тег имеет множество задач благодаря выбору его типа (**type**). Описания основных задач:

type	text	Ввод текстовой информации
	button	Кнопка
	hidden	Невидимые данные в виде текста
	checkbox	Выбор данных из группы данных
	radio	Выбор одних данных из группы данных
	file	Выбор файла (пути к файлу) с клиента
	password	Ввод пароля
	reset	Сброс введенных данных
	submit	Отправка данных

Разберем по ближе два типа `checkbox` и `radio`. Для объявления `checkbox`, `radio` нас появляется еще один дополнительный атрибут `checked="checked"`, он отвечает за выбранные данные. Рассмотрим пример с `checkbox`:



Для объявления типа **radio** нам понадобится добавить еще один атрибут **name** и создать несколько **<input/>** тегов, с одинаковыми именами (name). Смотрите пример:

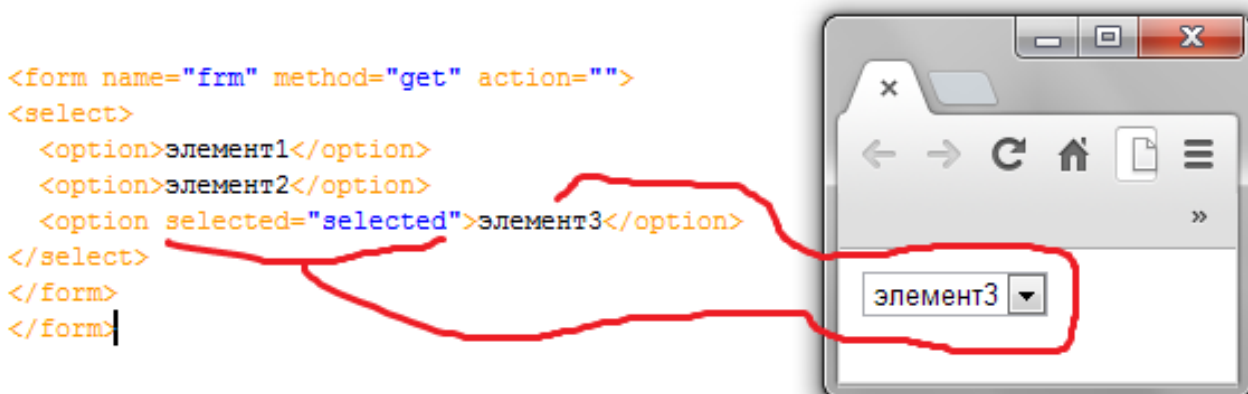


<button/>— имеет основные атрибуты: **type**, **value**. Он имеет три типа: button, reset и submit.

<select>/select>— Контейнер для создания списка выбора данных.

<option>/option>— имеет основной атрибут **selected="selected"**, предназначенный для выделения данных в меню. OPTION относится к SELECT'у для подразделения данных.

Пример объявления меню в форме:




```

<tr>
  <td>Логин</td>
  <td><input type="text" /></td>
</tr>
<tr>
  <td>Пароль</td>
  <td><input type="password" /></td>
</tr>
<tr>
  <td>Путь к фотографии</td>
  <td><input type="file" /></td>
</tr>
<tr>
  <td>Место проживания</td>
  <td>
    <select>
      <option>Калининград</option>
      <option>Светлогорск</option>
      <option>Светлый</option>
      <option>Гурьевск</option>
      <option>Гусев</option>
      <option>Советск</option>
    </select>
  </td>
</tr>
<tr>
  <td><input type="submit" /></td>
  <td><input type="reset" /></td>
</tr>

```

Содержание отчёта:

1. Наименование практического занятия.
2. Цель занятия.
3. Отчёт о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
4. Список использованных источников.
5. Выводы и предложения.

Вопросы для самопроверки:

1. Что является обязательными параметрами функции `fopen()`?
2. При помощи, какой функции можно записывать данные в файл при помощи PHP?
3. Что выступает в качестве обязательных параметров функции `fwrite()`?
4. Какую функцию используют для построчного считывания файла?
5. Сколько обязательных параметров имеет функция `fgets()`?
6. Какую функцию нужно использовать для того, чтобы считать файл как единое целое?

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 20/52

7. Возможно, ли содержимое сайта считать в массив? И если да то с помощью, какой функции?

8. Какая функция поможет определить размер файла?

Практическое занятие № 4 Использование изображений. Карты MAP

Цель занятия:

Закрепить навыки работы с изображениями.

Исходные материалы и данные.

1. Конспект лекций по теме
2. Персональный компьютер
3. Программа Notepad++

Содержание и порядок выполнения задания:

Карты MAP имеют механизм для клиентских карт изображений. Элементы MAP имеют имя, по которому он может быть отослан к элементу IMG, и содержит элементы AREA, которые определяют горячие зоны на связанном изображении и связывают эти горячие зоны с URL.

Визуальный элемент MAP никак не проявляется в документе, однако, вместе со связанными структурами он создает изображение реагирующей на нажатие клавиш карты.

Основной синтаксис:

<MAP>

<AREA элемент >

</MAP>

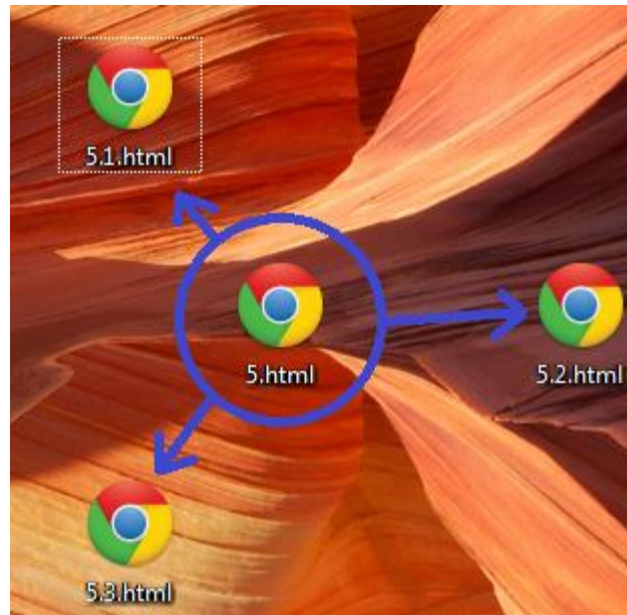
Атрибут MAP: NAME.

Атрибуты AREA: HREF - адрес, ALT-текстовое описание, SHAPE (=RECT(прямоугольник), CIRCLE(круг), POLY(многоугольник)) - очертание области, COORDS-координаты.

Атрибут : USEMAP

Задание:

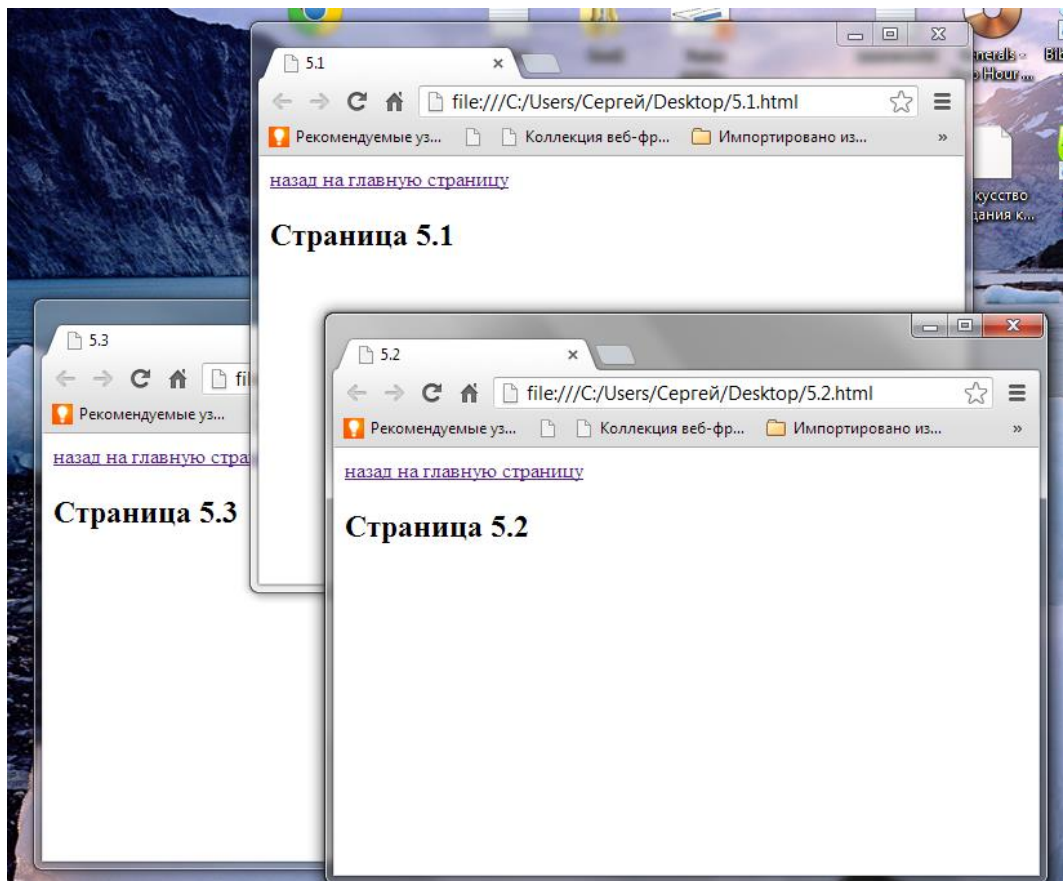
Для начала работы мы создадим несколько новых HTML документов. Одни из них будет главным. У вас должно получиться примерно так:



Откроем наш главный файл (5.html), и создадим там меню, которое будет ссылаться на другие файлы.

Так же отметим в других файлах (5.1.html;5.2.html;5.3html) заголовок с названием файла и ссылкой на главную страницу(5.html).

Примерно должно получиться так.



МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 22/52

Вернем к созданию меню. Откроем Paint, нарисуем такое изображение – меню (обязательно должны присутствовать многоугольник, окружность и прямоугольник.):



Приступим к написанию кода. Создадим 3 тега IMG(укажем путь к изображению и названия usemap), MAP(дадим название) и 3 – AREA(укажем параметры адрес и shape с координатами):

```

<map name="map1">
  <area href="5.1.html" shape="poly" coords="" />
  <area href="5.2.html" shape="circle" coords="" />
  <area href="5.3.html" shape="rect" coords="" />
</map>
```

Осталось только указать координаты. Снова откроем наше изображение и узнаем координаты многоугольника (звезды). Укажем их в строке `<area href="5.1.html" shape="POLY" coords="" />` coord. Так сделаем с кругом и прямоугольником.

```

<map name="map1">
  <area href="5.1.html" shape="poly" coords="103,7,130,48,181,48,159,94,182,140,132,140,103,184,74,140,25,140,47,94,25,50,74,48,103,7" />
  <area href="5.2.html" shape="circle" coords="201,320,131" />
  <area href="5.3.html" shape="rect" coords="294,54,447,168" />
</map>
```

Содержание отчёта:

1. Наименование практического занятия.
2. Цель занятия.
3. Отчёт о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 23/52

4. Список использованных источников.
5. Выводы и предложения.

Вопросы для самопроверки:

1. В чем заключается принцип работы кода `html special chars()` ?
2. Что является основным в работе с PHP?
3. Как можно преобразить `age` (возраст), о котором нам известно, что оно должно быть число?
4. Какие переменные автоматически установлены средствами PHP?

Практическое занятие №5 Пример использования таблиц стилей. Теги «div» и «span»

Цель занятия:

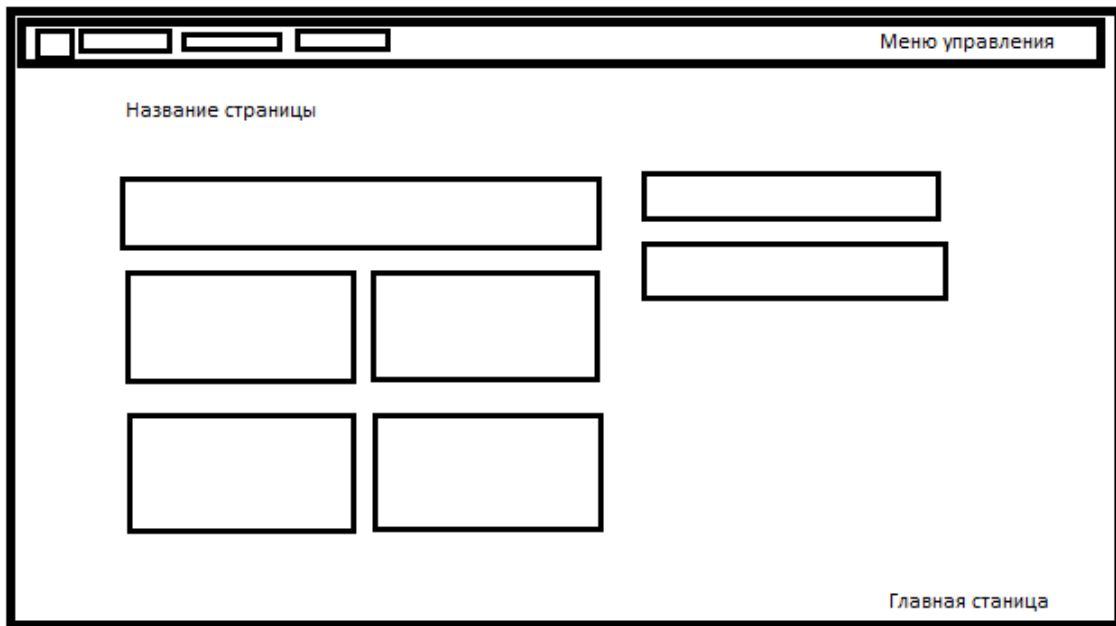
Закрепить навык работы с CSS

Исходные материалы и данные.

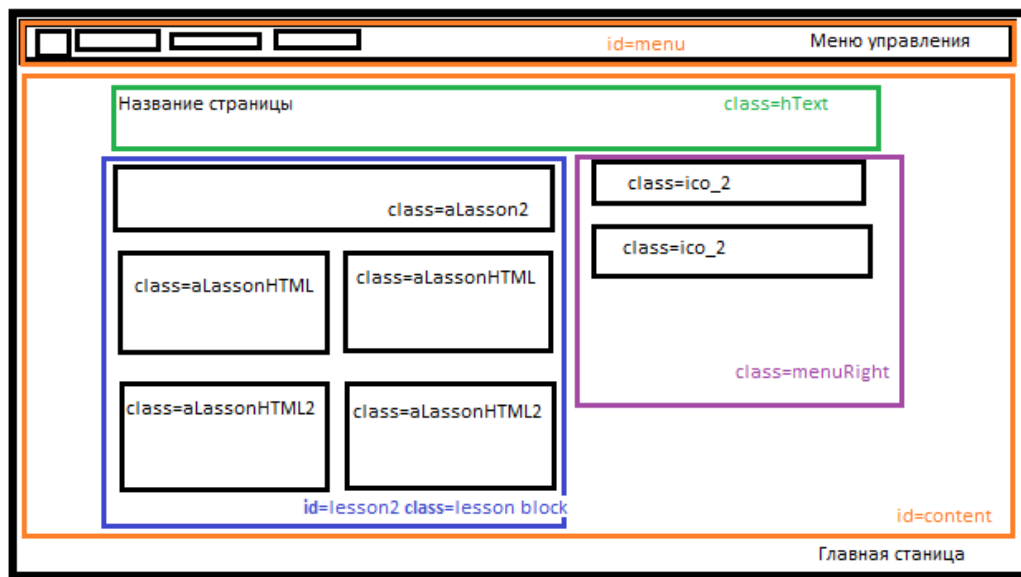
1. Конспект лекций по теме
2. Персональный компьютер
3. Программа Notepad++

Задание:

Прежде чем приступить к работе сделаем небольшой набросок нашей главной страницы. Вот сам набросок.



Теперь прикинем, из чего будет состоять наш сайт, разобьём на блоки.



Когда все расписали и продумали, подготовим изображения в паке img. Приступим к верстке сайта.CSS Сайта:

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 25/52

```
<style type="text/css">
@charset "windows-1251";
/* CSS Document */
body{
    margin:0 auto;
}
a{
    text-decoration:none;}
#menu{
    width:100%;
    height:30px;
    background:#00BCF2;
    padding:0 0 0 0;
    margin:0 auto;
}
#menu ul{
    margin:0 0 0 0;
    list-style-type:none;
    width:1000px;
}
#menu ul li{
    float:left;
    color:#fff;
    font-size:16px;
    cursor:pointer;
    margin:5px 0 0 20px;
}
#content{
    margin:0 auto;
    width:1000px;
}
.lesson{
    width:660px;
    min-height:400px;
    float:left;
}
.menuRight{
    float:left;
    width:340px;
    height:240px;
}
```

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 26/52

```

.hText{
    width:900px;
    height:36px;
    margin:30px 0 10px 0;
    padding:0 0 0 100px;
    color:#00BCF2;
    font-size:34px;
}
.aHome{
    display:block;
    width:34px;
    height:30px;
    background: url(img/aHome%201.png) no-repeat;
}
.aHome:hover{
    background:url(img/aHome%203.png) no-repeat;
}
.aLessonHTML, .aLessonHTML2, .aMenu, .aLesson2{
    color:#FFFFFF;
    float:left;
    margin:10px 10px 10px 10px;
    padding:0 20px 0 20px;
    display:block;
    width:260px;
    height:130px;
}
.aLesson1_top{
    font-size:23px;
    font-weight:bold;
    margin:10px 0 0 0;
}
.aLesson2{
    margin-right:441px;
    padding-left:50px;
    padding-top:40px;
    font-size:24px;
    width:559px;
    height:43px;
    background: url(img/logo_book_1.png) no-repeat;
}
.aLesson2:hover{
    background-position: 0px -117px;
}
/*-----Темно Синий блок-----*/
.aLessonHTML{
    background:#5C8DC2 url(img/blog.png) no-repeat 200px 70px;
    border:2px #88ABD2 solid;
}
.aLessonHTML:hover{
    border:2px #66CC33 solid;
    background:#5C8DC2 url(img/blog2.png) no-repeat 200px 70px;
}

```

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 27/52

```

/*-----Оранжевый блок-----*/
.aLessonHTML2{
    background:#FF9966 url(img/blog.png) no-repeat 200px 70px;
    border:2px #FFCC99 solid;
}
.aLessonHTML2:hover{

    border:2px #66CC33 solid;
    background:#FF9966 url(img/blog2.png) no-repeat 200px 70px;
}
/*-----*/
.ico{
    display:block;
    cursor:pointer;
    margin:5px 0 0 0;
    height:22px;
    background:url(img/link-list.png) no-repeat 155px 0;}
.ico_2{
    display:block;
    cursor:pointer;
    margin-top:10px;
    padding:10px 10px 10px 10px;
    color:#fff;
    width:338px;
    height:22px;
    border:#bfeded solid 2px;
    background:#6CC;
}

    .ico_2:hover{
        border:#6CC solid 2px;
    }
    .liHover{
        color:#fff;
        border-bottom:#fff 7px solid;
    }
    .li{
        color:#e8e8e8;
    }
    .li:hover{
        color:#fff;
    }
}
</style>

```

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 28/52

HTML код сайта:

```

<div id="menu">
  <ul>
    <li style="margin-top:0;"><a href="#0" class="aHome"></a></li>
    <li class="li">Раздел 1</li>
    <li class="liHover">Раздел 2</li>
  </ul>
</div>
<div id="content">
  <div class="hText">
    Название главной страницы
    <span style=" font-size:16px; color:#0072C6;">Главное меню</span>
  </div>
  <div id="lesson2" class="lesson block" style="display:block;">
    <a href="#1" class="aLesson2">
      Интересная ссылка
    </a>
    <a href="#2" class="aLessonHTML">
      <div class="aLesson1_top">Практическая работа 1</div>
      <b>Тема:</b> Название<br />
      <b>Описание:</b> описание
    </a>
    <a href="#3" class="aLessonHTML">
      <div class="aLesson1_top">Практическая работа 2</div>
      <b>Тема:</b> Название<br />
      <b>Описание:</b> описание
    </a>
    <a href="#4" class="aLessonHTML2">
      <div class="aLesson1_top">Практическая работа 3</div>
      <b>Тема:</b> Название<br />
      <b>Описание:</b> описание
    </a>
  </div>
  <div class="menuRight" style="display:block;">
    <a href="#5" class="ico_2" style="margin-top:32px;">Дополнительная ссылка 1</a>
    <a href="#6" class="ico_2">Дополнительная ссылка 2</a>
  </div>
</div>

```

Должен получиться такой результат:

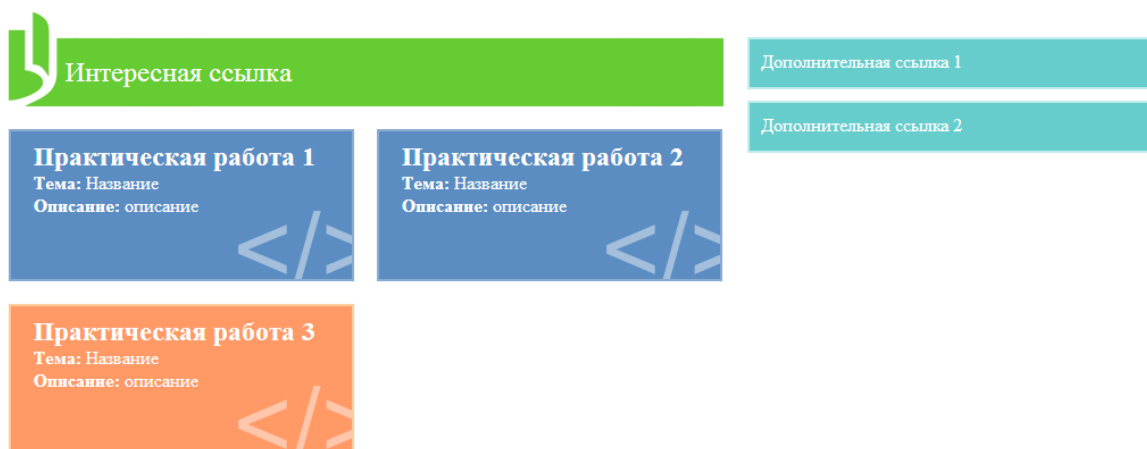
МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 29/52



Раздел 1

Раздел 2

Название главной страницы [Главное меню](#)



Содержание отчёта:

1. Наименование практического занятия.
2. Цель занятия.
3. Отчёт о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
4. Список использованных источников.
5. Выводы и предложения.

Вопросы для самопроверки:

1. Что задаётся тегом `$HOST='LOCALHOST';?`
2. Как расшифровывается данный запрос `$QUERY = "SELECT * FROM `MY_SQL_TABLE`";?`
3. Что делает запрос `$RES = MYSQL_QUERY($QUERY);?`

Практическое занятие № 6 Шрифты в CSS


Цель занятия: Закрепить навык преобразования шрифтов в CSS.

Исходные материалы и данные.

1. Конспект лекций по теме
2. Персональный компьютер
3. Программа Notepad++

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 30/52

Содержание и порядок выполнения задания:

Чтобы понимать, о чем идет речь, выберите любой шрифт со страницы Google Fonts и откройте его, нажав на кнопку Quick-use .

Шаг 1: выберите начертание

Первым делом на странице выбранного шрифта отобразятся варианты его начертания, а также иконка спидометра, которая означает не что иное, как скорость загрузки шрифта. Чем больше стилей для шрифта вы выбираете, тем больше времени потребуется на его загрузку. Поэтому рекомендуется выбирать только те варианты начертания, которые планируется использовать.

Шаг 2: выберите алфавит

Далее на странице есть возможность выбрать набор символов: латиница, кириллица и т. д. В зависимости от шрифта, в нем могут быть доступны не все варианты алфавита. Аналогично предыдущему пункту, лучше поставить галочку только напротив того алфавита, который понадобится.

Шаг 3: добавьте код на сайт

Далее Google предлагает подключить шрифт одним из нескольких способов: или стандартным, или через директиву @import, или с помощью JavaScript. Мы рассмотрим первые два варианта.

Первый способ подразумевает добавление в HTML-код ссылки на сервер Google, откуда и скачивается шрифт. Вам необходимо скопировать уже готовый кусок кода и поместить его между тегами <head></head> в вашем HTML-документе.

Пример:

```

1 <head>
2 <link href='http://fonts.googleapis.com/css?family=Roboto&subset=latin,cyrillic'
3 rel='stylesheet' type='text/css'>
4
5 ...
6 </head>
```

Второй способ – подключение шрифта с помощью директивы @import. Готовый код находится во второй вкладке пункта 3 на странице выбранного Google-шрифта. Его нужно добавить в самое начало вашей таблицы стилей (в противном случае файл не импортируется).

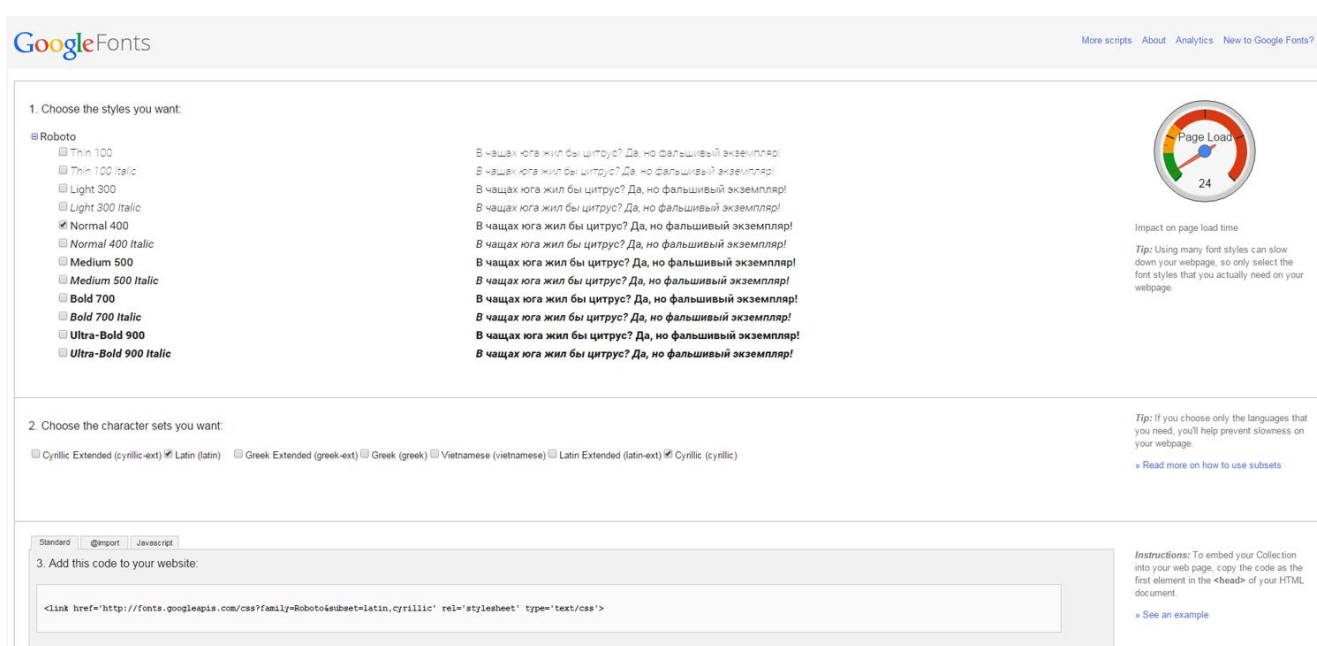
Выглядит код примерно так:

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 31/52

1 @import url(http://fonts.googleapis.com/css?family=Roboto&subset=latin,cyrillic);

Особенность первого способа заключается в том, что вам понадобится добавлять ссылку на шрифт в заголовок каждой страницы, где планируется его использовать. Это легко осуществить на сайтах с небольшим количеством страниц, но проблематично для крупных ресурсов. Второй способ удобен тем, что код можно поместить в самое начало внешней таблицы стилей, и тогда все страницы, к которым подключена эта таблица, получают необходимый шрифт, который будет загружаться с помощью директивы @import.



Скриншот: этапы настройки шрифта Google

Шаг 4: создайте стиль

После осуществления предыдущих шагов можно начать применять шрифт.

Как записывается такое правило CSS, вы уже видели ранее:

```
1 p {
2 font-family: 'Roboto', sans-serif;
3 }
```

Если на первом шаге вы выбрали несколько вариантов начертания (к примеру, добавили вариант жирного шрифта Bold 700), тогда на третьем шаге код немного видоизменится:

```
1 <link href='http://fonts.googleapis.com/css?family=Roboto:700,400&subset=latin,cyrillic'
2 rel='stylesheet' type='text/css'>
```

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 32/52

Или:

```
1 @import  
url(http://fonts.googleapis.com/css?family=Roboto:700,400&subset=latin,cyrillic);
```

Чтобы затем придать шрифту жирное начертание, запишите CSS-стиль таким образом:

```
1 p {  
2 font-family: 'Roboto', sans-serif;  
3 font-weight: 700;  
4 }
```

Примечание: в Google Fonts для обозначения насыщенности шрифта используются только условные единицы от 100 до 900. Так, нормальное начертание (по умолчанию) эквивалентно значению 400 (normal), а стандартное полужирное начертание эквивалентно 700 (bold).

Главными преимуществами сервиса Google Fonts является:

- легкость использования (разобраться с сервисом сможет даже новичок, а весь необходимый код генерируется автоматически – остается только скопировать его);
- доступность шрифтов (нет необходимости за них платить);
- предусмотрены все используемые форматы шрифтов (это означает, что каждый браузер сможет загрузить именно тот формат шрифта, с которым он работает).

Среди недостатков сервиса – не очень большое разнообразие шрифтов, особенно кириллических. К слову, в интернете есть и другие похожие сервисы, например, TypeKit (платный).

Содержание отчёта:

1. Наименование практического занятия.
2. Цель занятия.
3. Отчёт о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»

МО-09 02 07- МДК 05.02.06.ПЗ	КМПК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 33/52

4. Список использованных источников.
5. Выводы и предложения.

Вопросы для самопроверки:

1. Что делает запрос `$QUERY = "SELECT FIRSTNAME FROM `MY_SQL_TABLE`";`?
2. Что делает запрос `$QUERY = "SELECT SURNAME FROM `MY_SQL_TABLE` WHERE ID=5";`?
3. Что делает запрос `$QUERY = "ALTER TABLE `MY_SQL_TABLE` DROP `SURNAME`";`?

Практическое занятие №7 Поля в CSS. Списки в CSS

Цель занятия: Закрепить навык организации полей и списков в CSS.

Исходные материалы и данные.

1. Конспект лекций по теме
2. Персональный компьютер
3. Программа Notepad++

Содержание и порядок выполнения задания:

Двумя наиболее распространенными типами HTML-списков являются упорядоченный и неупорядоченный. Для их стилизации списков мы будем использовать свойство `list-style` CSS.

Ниже приведен пример неупорядоченного списка. В нем видно, что по умолчанию у списков используется выравнивание по левому краю:



`list-style-type`

Чтобы задать стиль маркера для пунктов списка, можно использовать свойство `list-style-type`.

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 34/52

Стандартное значение этого свойства – `disc`, но можно использовать и другие значения: `circle`, `square`, `none`. Кроме этого можно использовать такие значения, как `upper-alpha`, `lower-alpha`, `upper-roman`, `lower-roman`, `decimal` и т. д.

В приведенном ниже примере свойство `list-style-type` имеет значение `square`:

```
ul {  
    list-style-type : square;  
}
```



Если вам вообще не нужны маркеры, то CSS `list style type` необходимо присвоить значение `none`:

```
ul {  
    list-style-type : none;  
}
```



Поля и отступы

Чтобы добавить в список отступы, можно использовать свойство **margin**.

В приведенном ниже примере использования **CSS ul li list style** мы прибавляем поля по **30px** над и под списком:

```
ul {  
    list-style-type : square;  
    margin-top : 30px;  
    margin-bottom : 30px;  
}
```

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 35/52



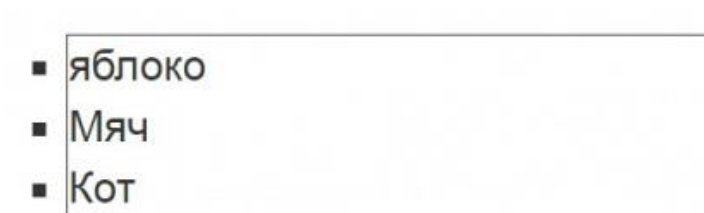
В следующем примере мы полностью исключаем поля:

```
ul {  
    list-style-type : square;  
    margin : 0;  
}
```



В примере CSS ul list style мы убираем отступы:

```
ul {  
    list-style-type : square;  
    margin : 0;  
    padding : 0;  
}
```



Здесь мы задаем списку фиксированную ширину:

```
ul {  
    list-style-type : square;  
    width : 200px;  
}
```

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 36/52

- Этот текст является элементом неупорядоченного списка.
- Это второй пункт того же списка.
- А это последний пункт.

List-style-position

Это свойство отвечает за то, как текст внутри **list style CSS** будет переходить на следующую строку. Стандартное значение **outside** означает, что при переносе строки текст также получит выравнивание по левому краю. Если выставлено значение **inside**, то строка будет перенесена под маркер списка.

В примере, приведенном ниже, представлен список, в котором свойство **list-style-position** выставлено на **outside** (значение по умолчанию):

```
ul {  
    list-style-type : square;  
    list-style-position : outside;  
    width : 200px;  
}
```

В следующем примере показано, что будет, если свойство **list-style-position** будет иметь значение **inside**. Таким образом, текст на новой строке будет начинаться прямо под маркером:

```
ul {  
    list-style-type : square;  
    list-style-position : inside;  
    width : 200px;  
}
```

- Этот текст является элементом неупорядоченного списка.
- Это второй пункт того же списка.
- А это последний пункт.

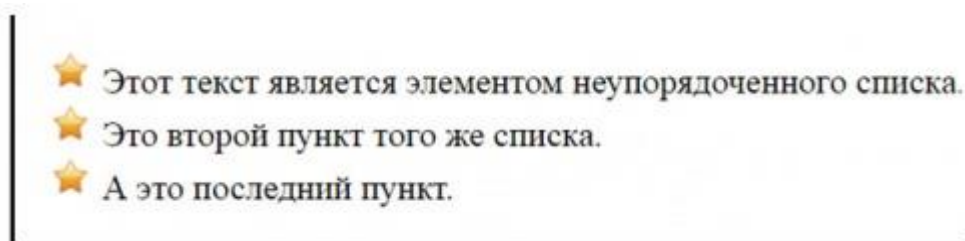
МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 37/52

List-style-image

Свойство **list-style-image** позволяет использовать собственную пиктограмму в качестве маркера.

В примере **CSS ul list style** мы используем изображение звезды вместо маркера списка:

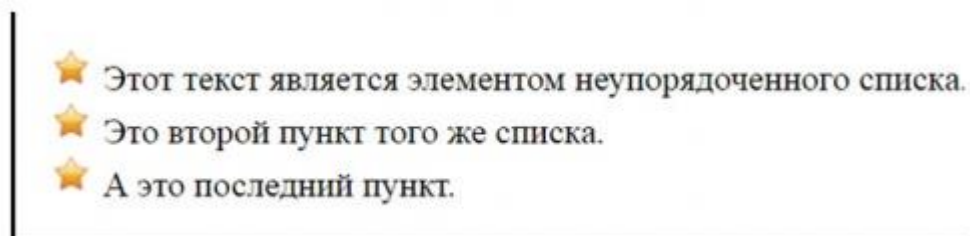
```
ul {  
    list-style-image: url(star.png);  
}
```



Сокращение list-style

Можно использовать сокращенное свойство **list-style** для назначения тех же стилей. В приведенном ниже примере мы задаем тип, расположение и изображение маркера списка:

```
ul {  
    list-style: square outside url(star.png);  
}
```



Список inline

По умолчанию, все пункты списка перечисляются с новой строки, то есть, вертикально. Если нужно выводить список горизонтально или строчно (**inline**), нужно присвоить элементам списка **display inline**.

В приведенном ниже примере представлен строчный список:

```
li {  
    display: inline;  
}
```



МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 38/52

Стилизация упорядоченного списка

Теперь пришло время задать стиль для **CSS ol list style**. По умолчанию, **list-style-type** для упорядоченного списка установлен при помощи десятичного числа:

1. яблоко
2. Мяч
3. Кот

Чтобы изменить тип, можно поменять значение на **upper-alpha**, **lower-alpha**, **upper-roman**, **lower-roman**.

Ниже приведен пример, где **list-style-type** имеет значение **upper-alpha**:

```
ol {  
    list-style-type : upper-alpha;  
}
```

- A. яблоко
- B. Мяч
- C. Кот

В следующем примере **list-style-type** имеет значение **lower-roman**:

```
ol {  
    list-style-type : lower-roman;  
}
```

- A. яблоко
- B. Мяч
- C. Кот

Содержание отчёта:

1. Наименование практического занятия.
2. Цель занятия.
3. Отчёт о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 39/52

4. Список использованных источников.
5. Выводы и предложения.

Вопросы для самопроверки:

1. Что делает запрос **list-style-type**?
2. Что делает запрос **ol list style**?

Практическое занятие № 8 Позиционирование в CSS

Цель занятия: Закрепить навык позиционирования в CSS.

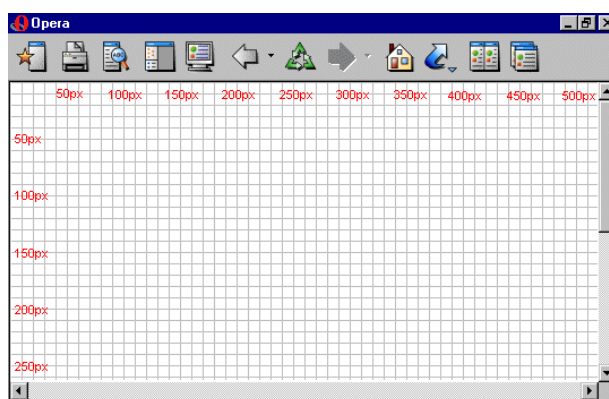
Исходные материалы и данные.

1. Конспект лекций по теме
2. Персональный компьютер
3. Программа Notepad++

Содержание и порядок выполнения задания:

При помощи CSS-позиционирования вы можете разместить, элемент точно в нужном месте страницы. Вместе с поплавками позиционирование даёт вам большие возможности для создания точного и навороченного дизайна.

Принципы CSS-позиционирования. Представим окно браузера как систему координат:



Принципы CSS-позиционирования - в том, что вы можете расположить бокс в системе координат где угодно.

Скажем, мы хотим позиционировать заголовок. При использовании боксовой модели заголовков выглядит так:

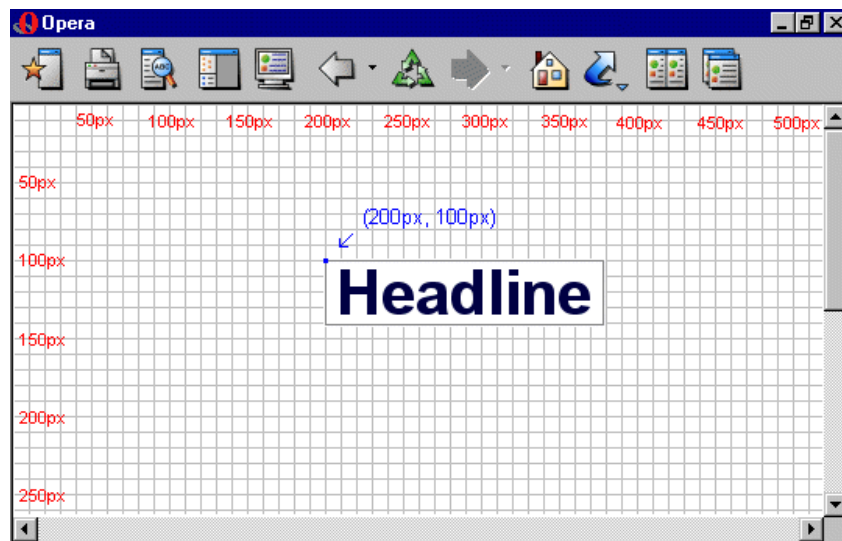
МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 40/52

Headline

Если мы хотим расположить его на 100px от верхней границы документа и на 200px слева, мы должны ввести следующий код CSS:

```
h1 {  
  position:absolute;  
  top: 100px;  
  left: 200px;  
}
```

Вот результат:



Как видите, Позиционирование с помощью CSS - очень точная техника при размещении элементов. Это намного проще, чем использовать таблицы, прозрачные изображения или ещё что-нибудь подобное.

Абсолютное позиционирование. Элемент, позиционированный абсолютно, не получает никакого пространства к документу. Это означает, что после позиционирования он не оставляет после себя пустое пространство.

Для абсолютного позиционирования элемента свойство *position* должно иметь значение *absolute*. Вы можете использовать значения *left*, *right*, *top* и *bottom* для размещения бокса.

```
#box1 {  
  position:absolute;  
  top: 50px;
```

*Документ управляется программными средствами 1С: Колледж
Проверь актуальность версии по оригиналу, хранящемуся в 1С: Колледж*

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 41/52

```
left: 50px;
}
```

Относительное позиционирование. Чтобы позиционировать элемент относительно, установите в свойстве *position* значение *relative*. Разница между относительным абсолютным позиционированием состоит в том, как обсчитывается позиционирование.

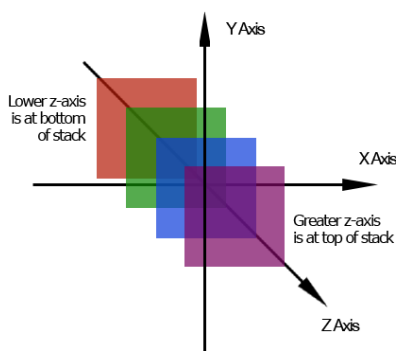
Позиция элемента, размещаемого относительно, обсчитывается относительно его оригинальной позиции в документе. Это означает, что вы смещаете элемент вправо, влево, вверх или вниз. Таким образом, элемент всё ещё занимает в документе пространство после позиционирования.

```
#dog1 {
position:relative;
left: 350px;
bottom: 150px;
}
```

Фиксированное позиционирование Fixed - По своему действию это значение близко к *absolute*, но в отличие от него привязывается к указанной свойствами *left*, *top*, *right* и *bottom* точке на экране и не меняет своего положения при прокрутке веб-страницы. Браузер Firefox вообще не отображает полосы прокрутки, если положение элемента задано фиксированным, и оно не помещается целиком в окно браузера. В браузере Opera хотя и показываются полосы прокрутки, но они никак не влияют на позицию элемента.

Static – Элементы отображаются как обычно. Использование свойств *left*, *top*, *right* и *bottom* не приводит к каким-либо результатам.

Z-index. Страница сайта двумерна. У неё есть ширина и высота. Z-index добавляет третье измерение, глубину.



МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 42/52

Чем выше этот индекс тем выше на странице расположен элемент. С помощью него мы можем добиться чтобы один элемент располагался поверх другого. По умолчанию, его значение - ноль. Отрицательные значения также допустимы.

На самом деле, z-index гораздо сложнее, чем я его здесь описываю, но это тема для отдельной статьи. Сейчас, главное запомнить саму идею третьего измерения и то, что только позиционируемые элементы могут использовать это свойство.

Задание 1:

Координаты слоя

Положение слоя в документе зависит от заданных значений стилевых свойств left, top, right и bottom, они устанавливают соответственно позицию слоя слева, сверху, справа и снизу. Точка отсчета определяется в зависимости от свойства position, которое принимает обычно значение relative (относительное положение) или absolute (абсолютное положение).

При абсолютном позиционировании слой располагается относительно родительского элемента или окна браузера, если родитель не задан, как показано на рис. 1.

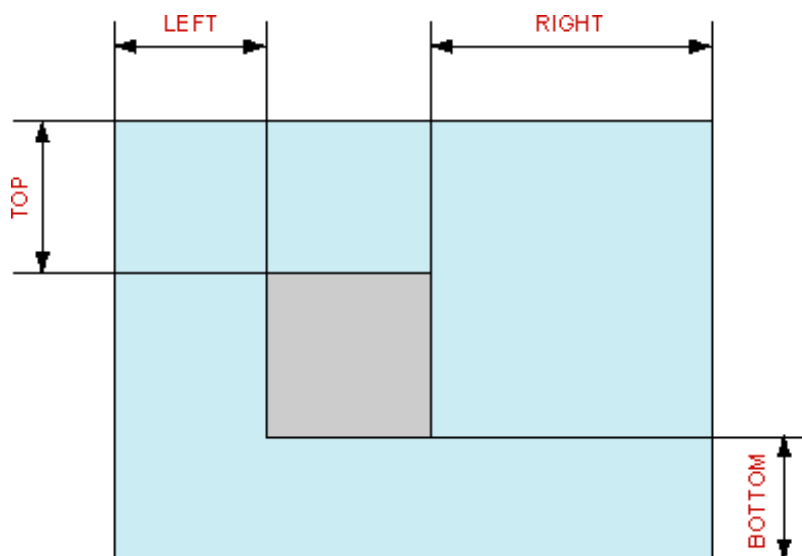


Рисунок 1 - Положение слоя относительно окна браузера

В случае использования относительного позиционирования, координаты отсчитываются от исходного положения слоя (рисунок. 2). Здесь пунктиром выделено первоначальное положение слоя, а сплошной линией результат его перемещения.

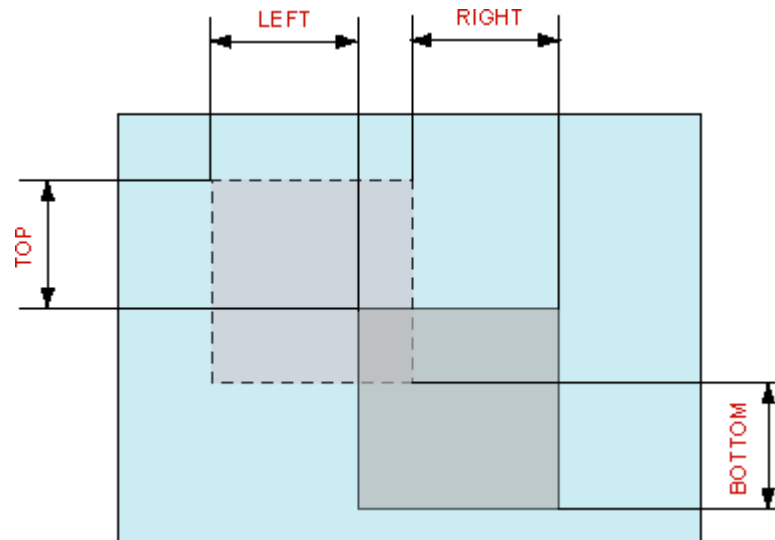


Рисунок 2. - Положение слоя относительно исходного положения

Положительные значения свойства `left` сдвигают слой вправо, отрицательные — влево. С `right` дело обстоит наоборот, положительные значения этого свойства смещают слой влево, а отрицательные — вправо. Аналогичная ситуация и с парой `top/bottom`. Значения со знаком минус у свойства `top` поднимают слой вверх, а положительные значения опускают слой вниз. `bottom` же действует с точностью наоборот.

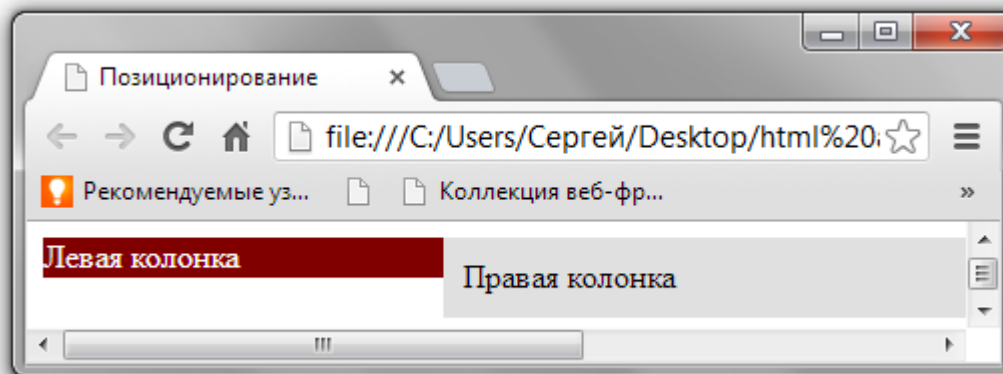
Заметим, что указание свойства `position` без упоминания координат (значений `left`, `top`, `right` или `bottom`) не меняет положение текущего слоя, но оказывает влияние на расположение близлежащих или вложенных слоев.

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 44/52

Пример 1. Размещение двух слоев

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Позиционирование</title>
<style type="text/css">
#leftcol {
position: absolute; /* Абсолютное позиционирование */
width: 200px; /* Ширина слоя */
background: #800000; /* Цвет фона */
color: white; /* Цвет текста */
}
#rightcol {
position: relative; /* Относительное позиционирование */
left: 200px; /* Смещаем слой вправо на ширину левого слоя */
width: 550px; /* Ширина слоя */
background: #e0e0e0; /* Цвет фона */
padding: 10px; /* Поля вокруг текста */
}
</style>
</head>
<body>
<div id="leftcol">Левая колонка</div>
<div id="rightcol">Правая колонка </div>
</body>
</html>
```

Результат:



Несмотря на простоту решения, в данном случае следует учесть несколько моментов.

Использование `position: absolute` для левой колонки, хотя и не дает видимого результата, необходимо для того, чтобы правая колонка не «перескакивала» вниз, и слои располагались на одном уровне.

Положение правого слоя `rightcol` зависит от значения стилевого свойства `left`, оно в данном случае должно равняться ширине левого слоя, в этом случае слои будут плотно прилегать между собой. Впрочем, ничего не мешает изменять этот

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 45/52

параметр в ту или иную сторону. Тогда между слоями появится разделительная полоса или наоборот, слои станут накладываться друг на друга.

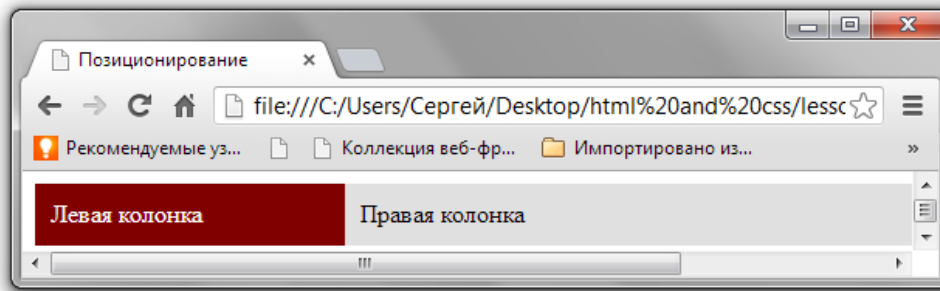
Свойство `padding`, добавляющее поля вокруг текста, если его использовать в стиле левого слоя `leftcol`, вносит искажения в макет страницы для браузера. Это связано с тем, что ширина элемента в браузере складывается из значений `width`, `padding` и `margin`. Поскольку добавление полей увеличивает ширину слоя, то правая колонка начинает накладываться поверх левой колонки. Чтобы устранить эту особенность, атрибут `padding` следует применить к стилю абзаца, который должен располагаться внутри слоя или изменить значение `width`. Также можно добавить еще один внутренний слой и указать поля для него (пример 2).

Пример 2. Поля внутри слоя.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Позиционирование</title>
<style type="text/css">
#leftcol {
    position: absolute; width: 200px; background: #800000; color: white;
}
#leftcol div {
    padding: 10px; /* Поля вокруг текста в левой колонке */
}
#rightcol {
    position: relative; left: 200px; width: 550px; background: #e0e0e0;
}
#rightcol div {
    padding: 10px; /* Поля вокруг текста в правой колонке */
}
</style>
</head>
<body>
<div id="leftcol">
    <div>Левая колонка</div>
</div>
<div id="rightcol">
    <div>Правая колонка</div>
</div>
</body>
</html>
```

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 46/52

Результат:



Задание 2 Повторите фиксированное и абсолютное позиционирование

Фиксированное позиционирование (fixed)

Задание элементу фиксированного позиционирования осуществляется посредством установки ему `position: fixed`.

Фиксированное позиционирование похоже на абсолютное, но в отличие от него оно всегда привязывается к краям окна браузера (viewport), и остаётся в таком положении даже при скроллинге страницы.

Фиксированное позиционирование применяется для закрепления на странице навигационных меню, кнопки «вверх», панелей с социальными кнопками и многого другого.



Абсолютное позиционирование (absolute)

Установка абсолютного позиционирования элементу осуществляется посредством задания ему `position: absolute`.

Этот тип позиционирования позволяет разместить элемент именно там, где вы хотите.

МО-09 02 07- МДК 05.02.06.ПЗ	КМПК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 47/52

Позиционирование выполняется относительно ближайшего позиционированного предка.

Под позиционированным элементом понимается элемент с position, равным relative, absolute, fixed или sticky.

Если среди предков у элемента с position: absolute нет позиционированного элемента, то в этом случае он будет позиционироваться относительно HTML страницы, т.е. элемента body.

Когда элементу устанавливаем position: absolute без указания CSS-свойств (top, left, right и bottom), определяющих его положение, он будет находиться в том месте, в котором он был бы расположен, если бы находился в потоке (при этом при вычислении его положения учитываются только элементы, расположенные до него в коде и находящиеся в потоке).

При этом другие элементы его видеть не будут, и, следовательно, они будут расположены на странице, не обращая никакого внимания на него.



CSS-свойства для управления положением абсолютно позиционированного элемента работают по-другому чем с position: relative.

CSS-свойства top, bottom, left и right задают положение элемента относительно ближайшего позиционированного предка или body, если такого предка нет.

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 48/52

Установить ширину (высоту) абсолютно позиционированному можно с помощью установки ему двух координат top и bottom (left и right).

Если элементу одновременно установить top, bottom и height, то предпочтение будет отдано top и height.

Абсолютное позиционирование применяется очень часто совместно с относительным позиционированием в дизайнерских целях, когда необходимо разместить различные элементы относительно друг друга, так же может применяться для создания выпадающих меню, разметки сайта и т.д.



Содержание отчёта:

1. Наименование практического занятия.
2. Цель занятия.
3. Отчёт о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
4. Список использованных источников.
5. Выводы и предложения.

Вопросы для самопроверки:

1. Опишите принципы позиционирования
2. Опишите фиксированное позиционирование Fixed

Практическое занятие № 9 Контекстные селекторы

Цель занятия: Изучить применение контекстных селекторов.

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 49/52

Исходные материалы и данные.

1. Конспект лекций по теме
2. Персональный компьютер
3. Программа Notepad++

Содержание и порядок выполнения задания:

Селектор может состоять из нескольких частей, разделённых пробелом, например:

```
p strong { ... }
ul .hit { ... }
.footer .menu a { ... }
```

Такие селекторы называют контекстными или вложенными. Их используют для того, чтобы применить стили к элементу, только если он вложен в нужный элемент.

Например, селектор `.menu a` сработает для ссылки `a` только в том случае, если она расположена внутри элемента с классом `.menu`.

Читать их проще всего справа налево:

```
/* выбрать все теги strong внутри тегов p */
p strong { ... }

/* выбрать все элементы с классом .hit внутри тегов ul */
ul .hit { ... }

/* выбрать все ссылки внутри элементов с классом .menu, которые лежат внутри элементов
с классом .footer */
.footer .menu a { ... }
```

Таким образом, можно задавать элементам различные стили в зависимости от их контекста. Если ссылка расположена внутри меню, сделать её крупнее, а если внутри основного текста, то задать ей нужный цвет.

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 50/52

Соседние селекторы

Контекстные селекторы используются для вложенных друг в друга элементов, а соседние — для расположенных рядом.

Например, теги `` в списке являются соседними по отношению друг к другу и вложенными в тег ``.

Соседние селекторы записываются с помощью знака `+`, например, селектор `1 + селектор2`. Стили применяются к элементу, подходящему под селектор2, только если сразу перед ним расположен элемент, подходящий под селектор1.

Пример. Есть два элемента списка:

```
<ul>
  <li class="hit"></li>
  <li class="miss"></li>
</ul>
```

Селектор `.hit + .miss` применит стили к элементу с классом `miss`, так как перед ним есть элемент с классом `hit`.

Селектор `.hit + li`, а также селектор `li + .miss`, или даже `li + li` тоже применит стили к элементу с классом `miss`, то есть ко второму элементу списка.

А вот селектор `.miss+.hit` не работает, так как элемент с классом `miss` находится после элемента с классом `hit` в разметке.

Контекстные и соседние селекторы

Селекторы в CSS можно очень гибко комбинировать. В частности, можно комбинировать контекстные и соседние селекторы.

Например, селектор `.player-1 .hit + .miss` работает для тега с классом `miss`, если сразу перед ним расположен тег с классом `hit` и оба тега расположены внутри тега с классом `player-1`.

Задания

1. Создайте произвольный текст с заголовками и несколькими абзацами
2. Создадим и закрасим содержимое тега **div** в красный цвет (используя атрибут **style**):
3. Создадим и закрасим содержимое тега **span** в зеленый цвет
4. Сгруппируйте теги заголовков и измените их цвет

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 51/52

5. Создайте атрибут **id** и закрасьте содержимое синим цветом

6. Бывают такие ситуации, когда разные теги имеют один и тот же класс. К примеру, у абзацев и заголовков h2 одновременно задан класс **test**. Вам может потребоваться выбрать **только абзацы** с классом **test**, не затрагивая заголовков **h2** с этим же классом.

В этом случае вместо селектора **.test** я должен написать следующее: **p.test** - таким образом я выберу все абзацы с классом **test**, не затронув заголовки. Если же я напишу **h2.test** - то выберу все заголовки h2 с классом **test**, не затронув абзацев.

Создайте в своем документе такой блок

Содержание отчёта:

1. Наименование практического занятия.
2. Цель занятия.
3. Отчёт о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
4. Список использованных источников.
5. Выводы и предложения.

Вопросы для самопроверки:

1. Как записываются соседние селекторы?
2. Что такое селектор?

Практическое занятие №10 Работа над html страницей по индивидуальному заданию

Цель занятия: Применить полученные знания на практике

Исходные материалы и данные.

1. Конспект лекций
2. Персональный компьютер
3. Программа Notepad++

Содержание и порядок выполнения задания:

Применить полученные знания на практике. Разработать веб-страничку со следующими обязательными элементами:

МО-09 02 07- МДК 05.02.06.ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ ПРОГРАММИРОВАНИЕ	С. 52/52

Заголовки

Текст с оформлением различными шрифтами

Формы (можно для получения отзывов или опрос)

Таблица (возможна для позиционирования)

Ссылки

Списки (различных форматов)

Картинки

Примерная тематика странички:

1. Историческое событие
2. Магазин
3. Рассказ о любимом животном
4. Персонажи компьютерных игр
5. Бюро путешествий
6. Страница о любимом (или популярном) музыканте или группе

Содержание отчёта:

1. Наименование практического занятия.
2. Цель занятия.
3. Вариант
4. Отчёт о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
5. Список использованных источников.
6. Выводы и предложения.

Вопросы для самопроверки:

1. Для чего нужен html?
2. Для чего нужен css?
3. Какие еще языки для гипертекста вы знаете?