



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

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

ПМ.05. МДК.05.02.06 Тема 5.2.6.1 ПЗ Интернет-программирование

Методические указания для выполнения практических занятий
по специальности

09.02.07 Информационные системы и программирование

МО–09 02 07- ПМ.05. МДК.05.02.06 Тема 5.2.6.1 ПЗ

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

МО – 09 02 07-ПМ.05. МДК.05.02.06 Тема 5.2.6.1 ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ-ТЕХНОЛОГИИ	С. 2/46

Содержание

ВВЕДЕНИЕ	3
ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ.....	4
Практическое занятие №1 Применение гипертекстовых ссылок	5
Практическое занятие № 2 Построение таблиц различного вида	8
Практическое занятие № 3. Формы ввода данных	16
Практическое занятие № 4 Использование изображений. Карты MAP	20
Практическое занятие № 5 Пример использования таблиц стилей.....	23
Практическое занятие № 6 Шрифты в CSS	29
Практическое занятие № 7 Поля в CSS. Списки в CSS	33
Практическое занятие № 8 Позиционирование в CSS.....	38

МО – 09 02 07-ПМ.05. МДК.05.02.06 Тема 5.2.6.1 ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ-ТЕХНОЛОГИИ	С. 3/46

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 1.1.1 Интернет программирование по теме «Введение в HTML и CSS» и предусматривает проведение 8 практических занятий-16 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 заданию с выводами по работе, должен знать теорию по данной теме, уметь пояснить, как проводилась работа и уметь анализировать полученные результаты.

МО – 09 02 07-ПМ.05. МДК.05.02.06 Тема 5.2.6.1 ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ-ТЕХНОЛОГИИ	С. 4/46

ПЕРЕЧЕНЬ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

№ п/п	Наименование практических занятий	Кол-во часов
1	Применение гипертекстовых ссылок.	2
2	Построение таблиц различного вида.	2
3	Формы ввода данных	2
4	Использование изображений. Карты MAP.	2
5	Пример использования таблиц стилей.	2
6	Шрифты в CSS.	2
7	Поля в CSS. Списки в CSS.	2
8	Позиционирование в CSS.	2
Итого		16

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.).

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

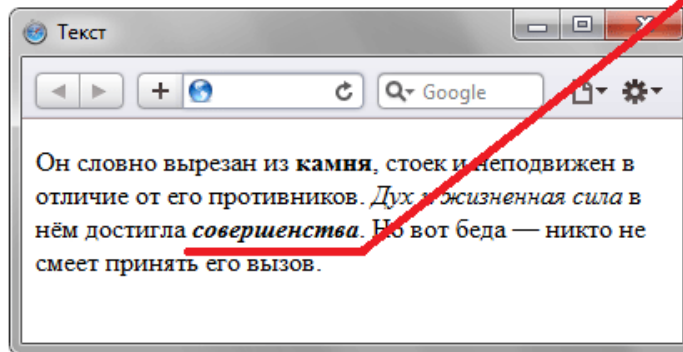


Рисунок 1.1.

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.

5. Отформатируем текст, таким образом:

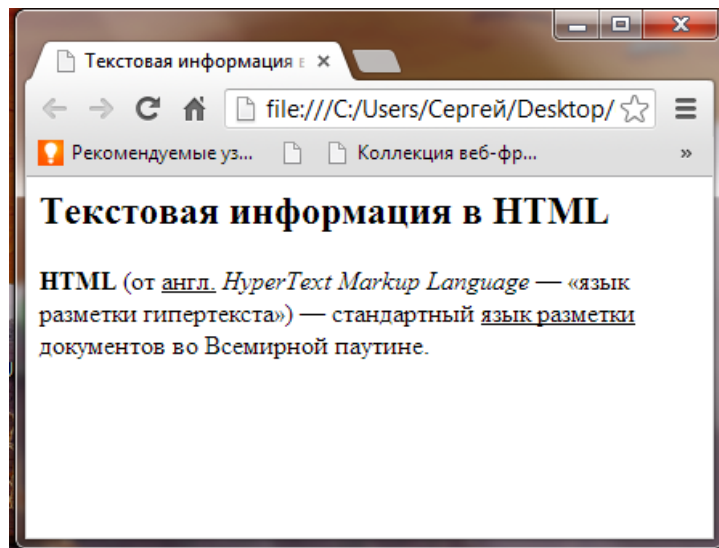


Рисунок 1.3.

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 гидроксилamina 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,

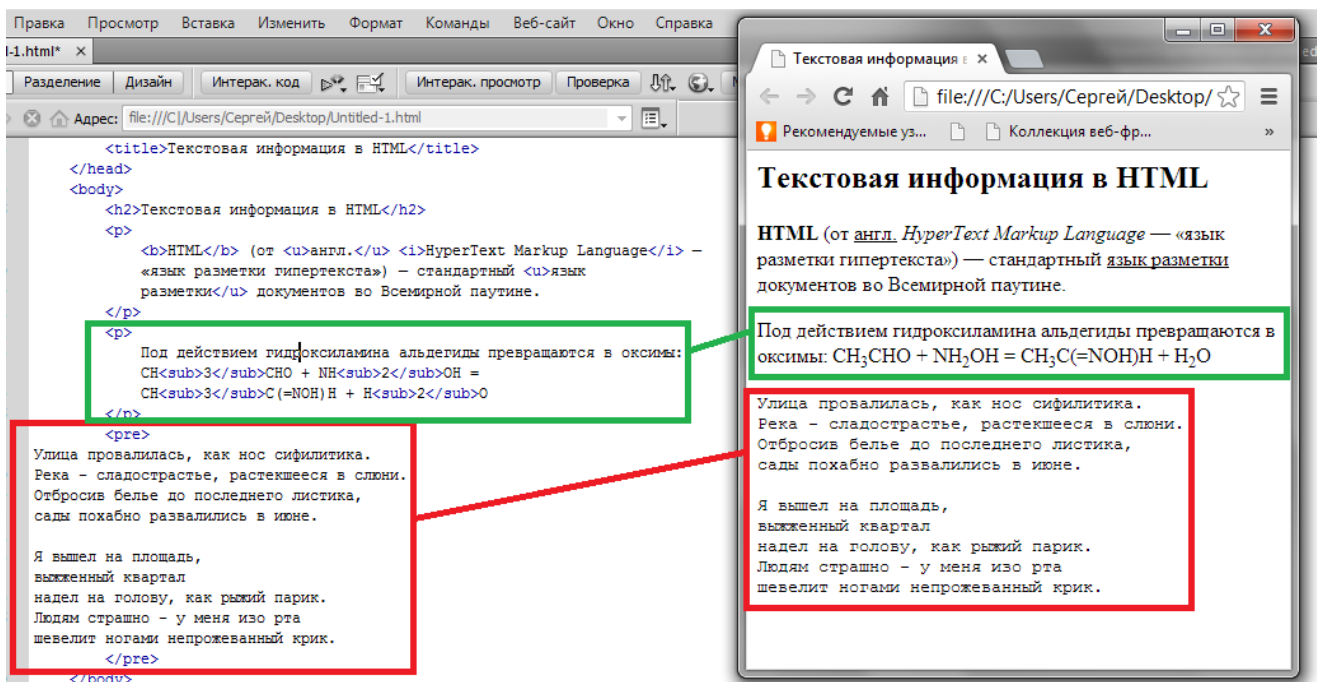
выжженный квартал

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

Цель занятия:

Научиться строить таблицы различного вида

Исходные материалы и данные.

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:

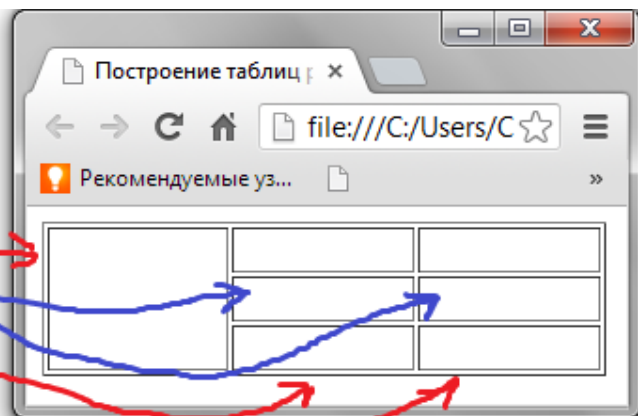
Сложная таблица:

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:

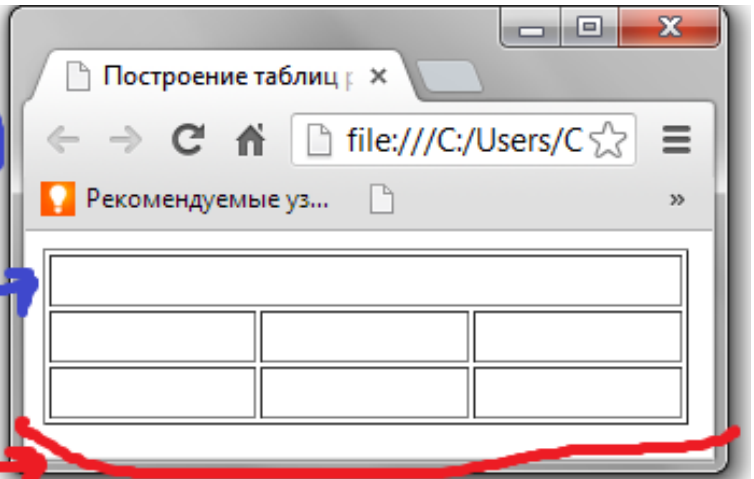
```
<table width="300" border="1">
  <tr>
    <td rowspan=3>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```



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		Кол-во контрольных работ	Количество часов			Распределение по курсам и предметам								1	
		Экзаменов	Курс. проект		Всего	Из них	2 курс		3 курс		4 курс		5 курс		2		
					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.	3
																4	
																5	
																6	
																7	
																8	
Факультативные предметы, консультации	Всего	Изучаемых предметов															9
		Курсовых проектов															10
		Экзаменов															11

В нашем случае строк будет 10, а столбцов 17.

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

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

№ п/п	Наименование предмета	Распределение по предметам		Кол-во контрольных работ	Количество часов			Распределение по курсам и предметам										
					Всего	Из них			2 курс		3 курс		4 курс		5 курс			
		Экзаменов	Курс. проект	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.			
	Факультативные предметы, консультации		Всего	Исучаемых предметов														
				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:

```

<table BORDER Width=100% >
<tr>
<td ROWSPAN=3 VALIGN=TOP>№ п/п</td>
<td ROWSPAN=3 VALIGN=TOP>Наименование предмета</td>
<td ROWSPAN=2 COLSPAN=2>Распределение по предметам</td>
<td ROWSPAN=3>Кол-во контрольных работ</td>
<td COLSPAN=4>Количество часов</td>
<td COLSPAN=8>Распределение по курсам и предметам</td>
</tr>
<tr>
<td ROWSPAN=2>Всего</td>
<td COLSPAN=3>Из них</td>
<td COLSPAN=2>2 курс</td>
<td COLSPAN=2>3 курс</td>
<td COLSPAN=2>4 курс</td>
<td COLSPAN=2>5 курс</td>
</tr>

```



```
<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>
<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>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<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:

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

МО – 09 02 07-ПМ.05. МДК.05.02.06 Тема 5.2.6.1 ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ-ТЕХНОЛОГИИ	С. 15/46

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

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>**.

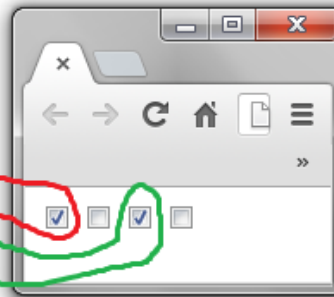
Рассмотрим по порядку теги.

<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

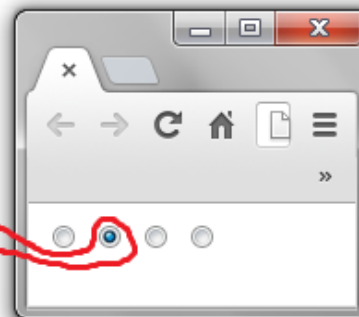
Разберем по ближе два типа *checkbox* и *radio*. Для объявления *checkbox*, *radio* нас появляется еще один дополнительный атрибут **checked="checked"**, он отвечает за выбранные данные. Рассмотрим пример с *checkbox*:


```
<form name="frm" method="get" action="">  
<input type="checkbox" checked="checked" />  
<input type="checkbox" />  
<input type="checkbox" checked="checked" />  
<input type="checkbox" />  
  
</form>
```



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

```
<form name="frm" method="get" action="">  
<input name="r1" type="radio" />  
<input name="r1" type="radio" checked="checked" />  
<input name="r1" type="radio" />  
<input name="r1" type="radio" />  
  
</form>
```



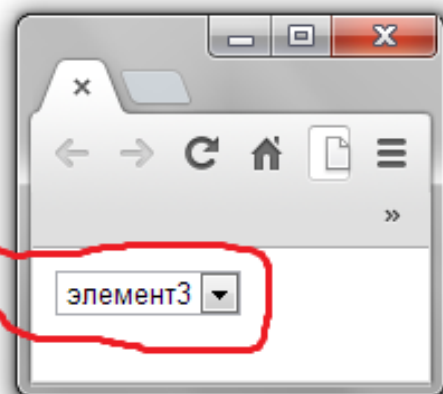
<button/>– имеет основные атрибуты: **type**, **value**. Он имеет три типа: button, reset и submit.

<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:

```
<form name="frm" method="get" action="">  
<select>  
  <option>элемент1</option>  
  <option>элемент2</option>  
  <option selected="selected">элемент3</option>  
</select>  
</form>  
</form>
```



Задания:

Для начала надо создать HTML-документ. Мы создадим такую форму:

Фамилия	<input type="text"/>
Имя	<input type="text"/>
Отчество	<input type="text"/>
Логин	<input type="text"/>
Пароль	<input type="password"/>
Путь к фотографии	<input type="text"/>
Место проживания	<input type="text"/>
	<input type="checkbox"/>
<input type="button" value="Отправить"/>	<input type="button" value="Сбросить"/>

Создадим форму, и чтобы наша форма выглядела ровно, создадим в форме таблицу из 2 столбцов и 8 строк с BORDER=0.

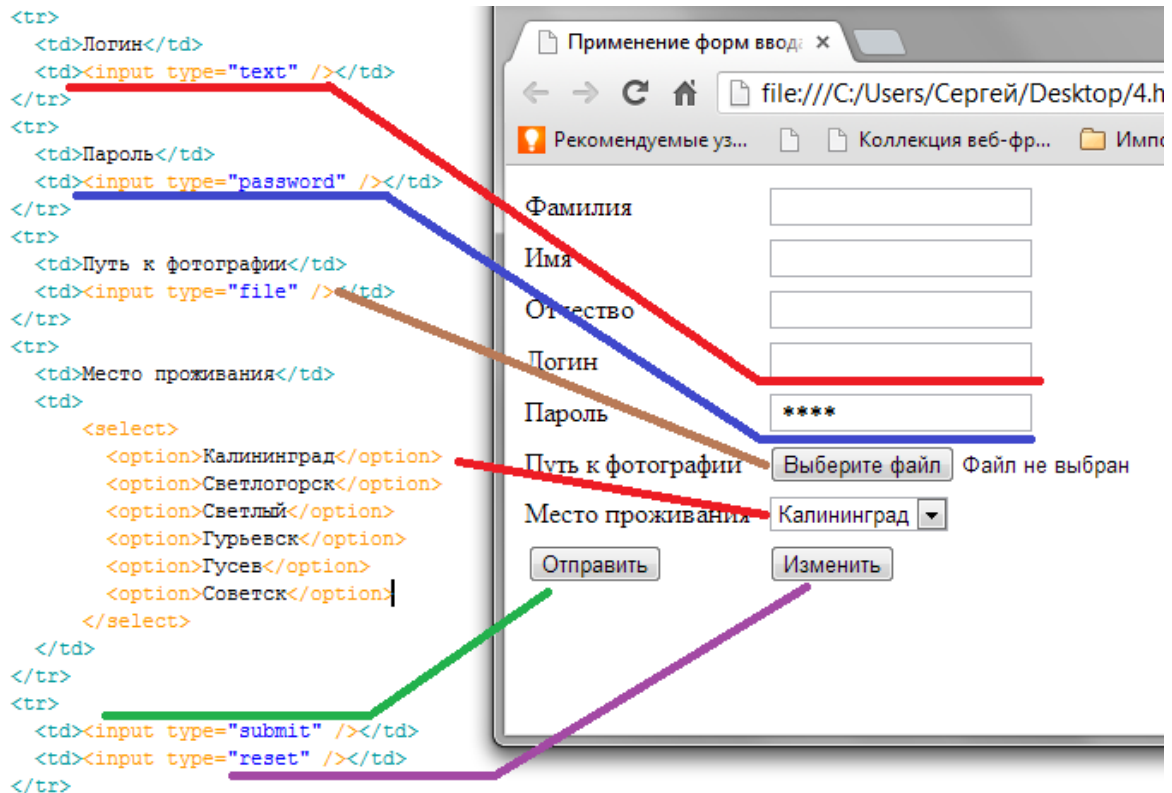
Вот что должно получиться:

```
<form>
<table width="200" border="0">
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
```

И далее таблица следует

Создадим нужные нам теги в нужном месте с нужным типом, а это `<input/>`, `<select></select>`, `<option></option>`.

Наш результат с кодом:



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

МО – 09 02 07-ПМ.05. МДК.05.02.06 Тема 5.2.6.1 ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ-ТЕХНОЛОГИИ	С. 20/46

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 содержат 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:

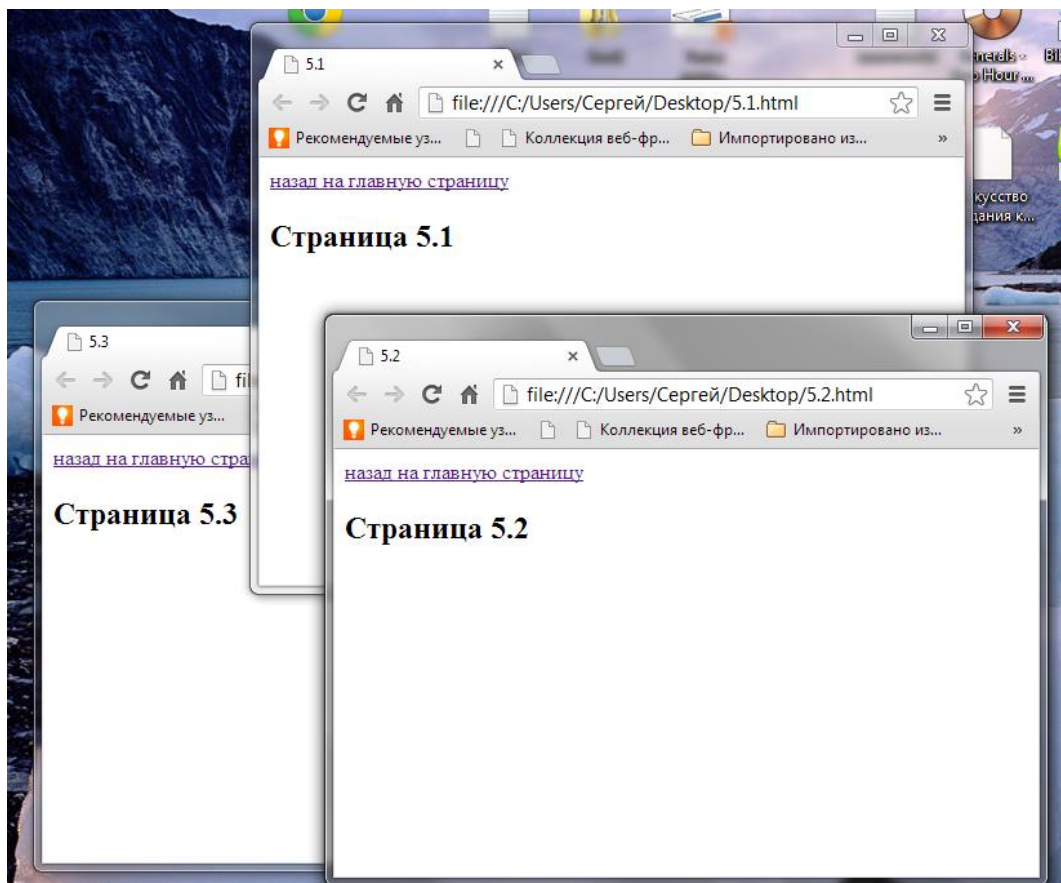
Для начала работы мы создадим несколько новых HTML документов. Одни из них будет главным. У вас должно получиться примерно так:



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.



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 shapес 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»

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

Цель занятия:

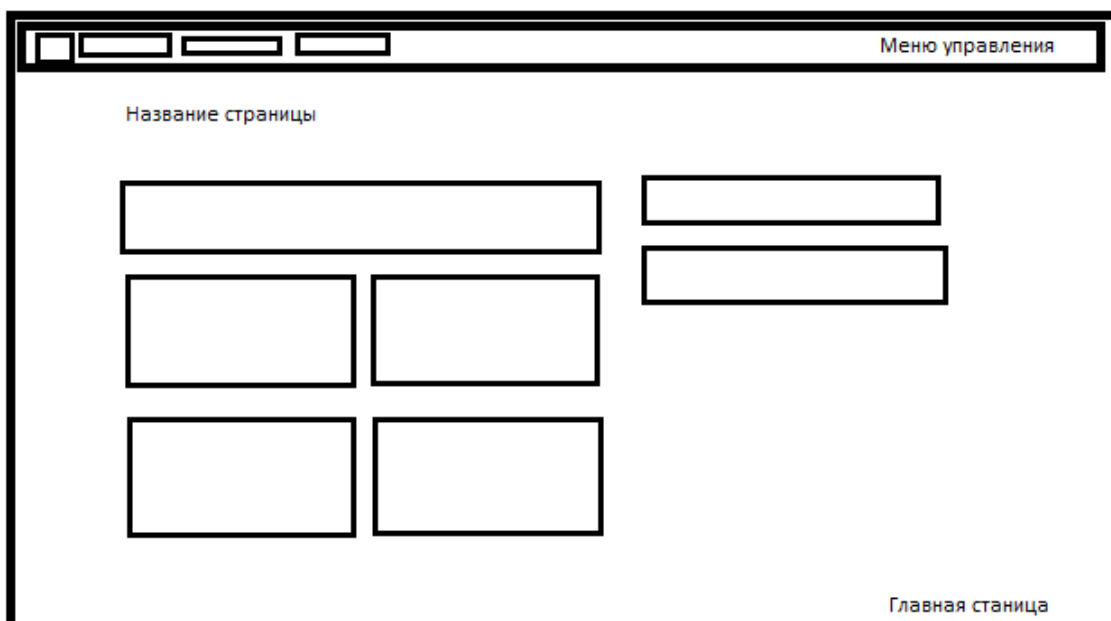
Закрепить навык работы с CSS

Исходные материалы и данные.

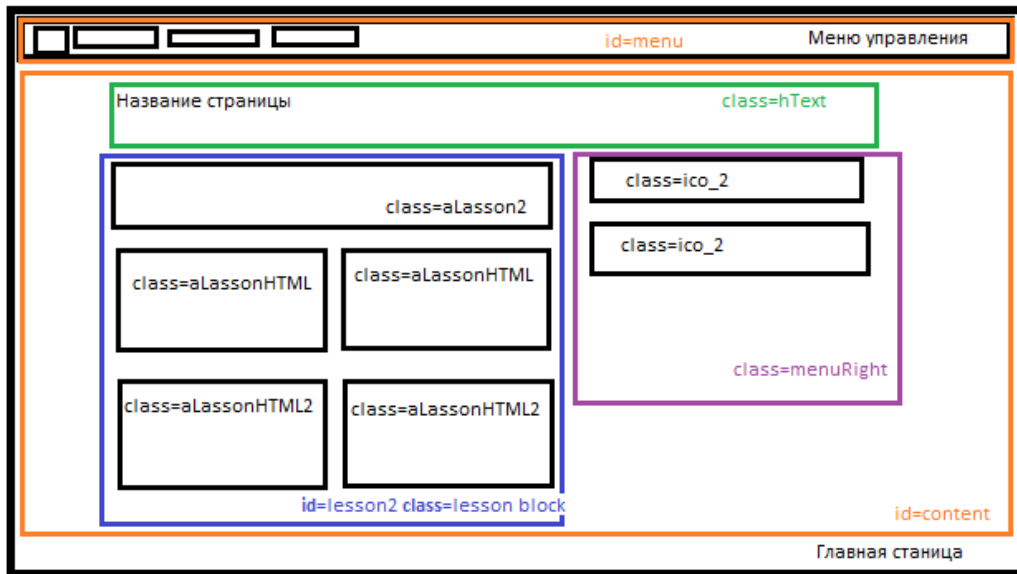
1. Конспект лекций по теме
2. Персональный компьютер
3. Программа Notepad++

Задание:

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



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



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

```
<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;
}
```

```
.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;
}
```

```
/*-----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>
```

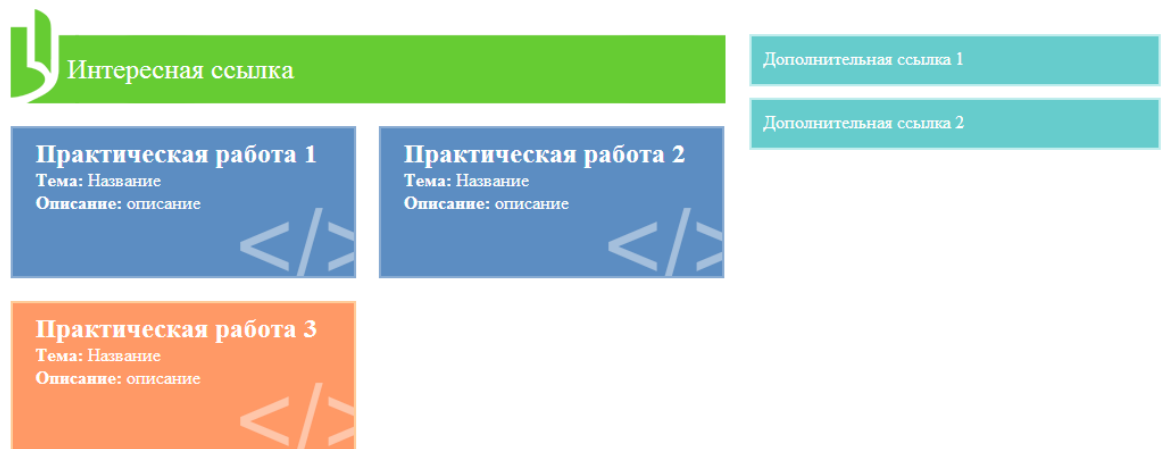
HTML код сайта:

```
<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:



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++

Содержание и порядок выполнения задания:

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

МО – 09 02 07-ПМ.05. МДК.05.02.06 Тема 5.2.6.1 ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ-ТЕХНОЛОГИИ	С. 30/46

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 планируются 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:

```
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, получат необходимый шрифт, который будет загружаться с помощью директивы `@import`.

The screenshot shows the Google Fonts website interface. At the top, there's a 'Page Load' gauge showing a score of 24. Below it, a tip states: 'Impact on page load time. Tip: Using many font styles can slow down your webpage, so only select the font styles that you actually need on your webpage.' The main content area is divided into three steps:

- 1. Choose the styles you want:** A list of Roboto font styles is shown, with 'Normal 400' selected. To the right, there are several lines of text: 'В чащах юга жил бы цитрус? Да, но фальшивый экземпляр!' repeated for different font weights.
- 2. Choose the character sets you want:** Radio buttons for 'Cyrillic Extended (cyrillic-ext)', 'Latin (latin)', 'Greek Extended (greek-ext)', 'Greek (greek)', 'Vietnamese (vietnamese)', 'Latin Extended (latin-ext)', and 'Cyrillic (cyrillic)'. 'Latin (latin)' and 'Cyrillic (cyrillic)' are selected. A tip says: 'Tip: If you choose only the languages that you need, you'll help prevent slowness on your webpage. > Read more on how to use subsets'.
- 3. Add this code to your website:** A code editor shows the following code: `<link href='http://fonts.googleapis.com/css?family=Roboto&subset=latin,cyrillic' rel='stylesheet' type='text/css'>`. A tip says: 'Instructions: To embed your Collection into your web page, copy the code as the first element in the <head> of your HTML document. > See an example'.

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'>
```

Или:

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

МО – 09 02 07-ПМ.05. МДК.05.02.06 Тема 5.2.6.1 ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ-ТЕХНОЛОГИИ	С. 32/46

```

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

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

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`.

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;  
}
```

- яблоко
- Мяч
- Кот

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

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

- яблоко
- Мяч
- Кот

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;  
}
```

- яблоко
- Мяч
- Кот

В следующем примере мы полностью исключаем поля:

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

- яблоко
- Мяч
- Кот

В примере CSS ul list style мы убираем отступы:

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

- яблоко
- Мяч
- Кот

Здесь мы задаем списку фиксированную ширину:

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

- 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

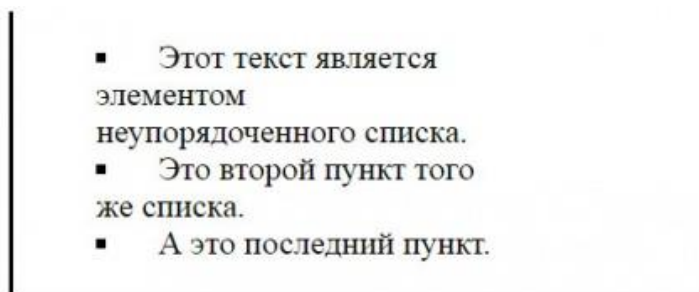
строки текст также получит выравнивание по левому краю. Если выставлено значение **inside**, то строка будет перенесена под маркер списка.

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;  
}
```



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);  
}
```

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

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);  
}
```

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

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;  
}
```

яблоко Мяч Кот

Стилизация упорядоченного списка

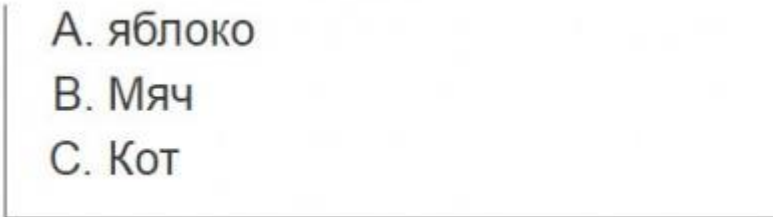
Теперь пришло время задать стиль для **CSS ol list style**. По умолчанию, **list-style-type** для упорядоченного списка установлен при помощи десятичного числа:

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**:

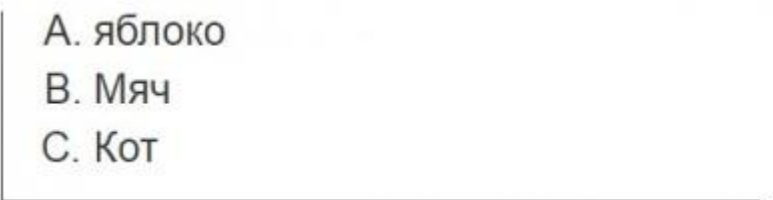
```
ol {  
    list-style-type : upper-alpha;  
}
```



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»
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.

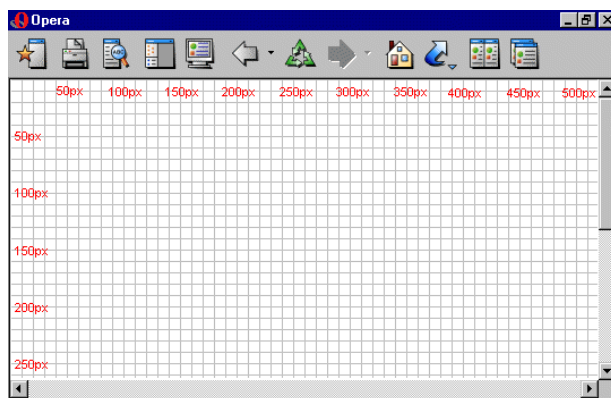
Исходные материалы и данные.

4. Конспект лекций по теме
5. Персональный компьютер
6. 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:

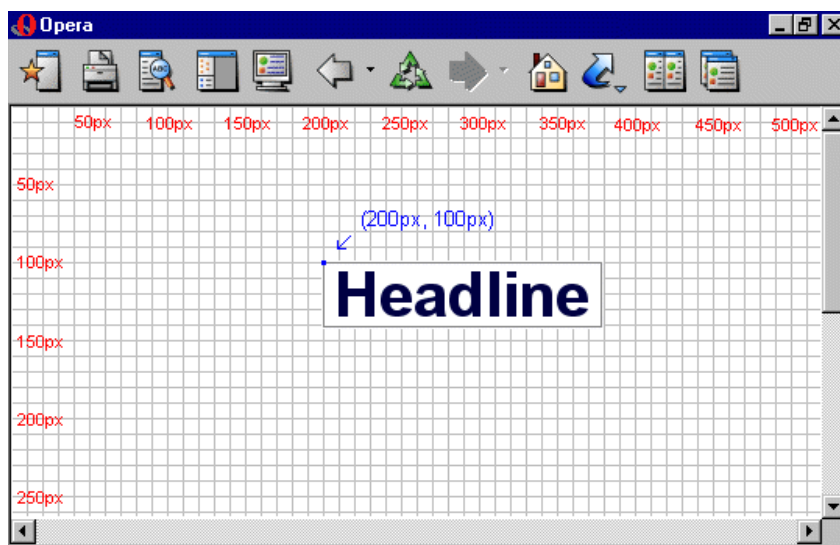
Headline

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 блока.

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

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

Позиция элемента, размещаемого относительно, обчисляется относительно его оригинальной позиции в документе. Это означает, что вы смещаете элемент

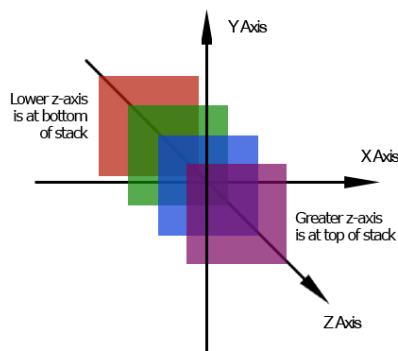
вправо, влево, вверх или вниз. Таким образом, элемент всё ещё занимает в документе пространство после позиционирования.

```
#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.



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:

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

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.

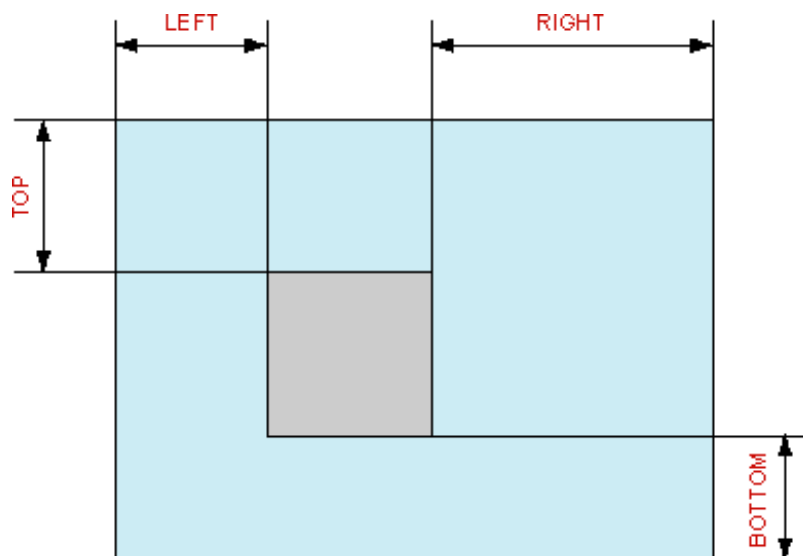
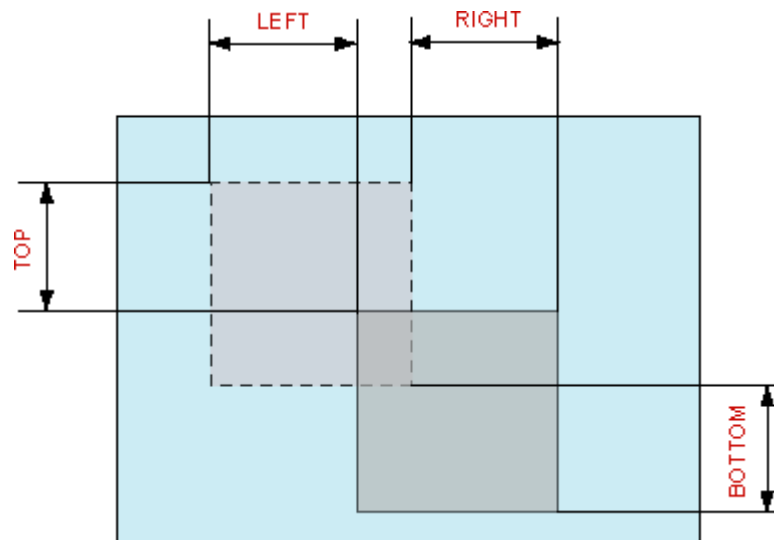


Рисунок 1 - Положение слоя относительно окна браузера

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.



МО – 09 02 07-ПМ.05. МДК.05.02.06 Тема 5.2.6.1 ПЗ	КМРК БГАРФ ФГБОУ ВО «КГТУ»	
	ИНТЕРНЕТ-ТЕХНОЛОГИИ	С. 43/46

Рисунок 2. - Положение слоя относительно исходного положения

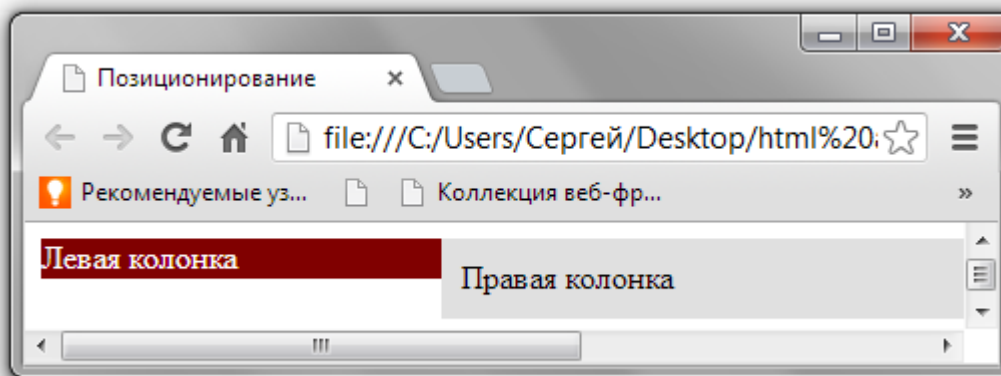
Положительные значения свойства `left` сдвигают слой вправо, отрицательные — влево. С `right` дело обстоит наоборот, положительные значения этого свойства смещают слой влево, а отрицательные — вправо. Аналогичная ситуация и с парой `top/bottom`. Значения со знаком минус у свойства `top` поднимают слой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

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.

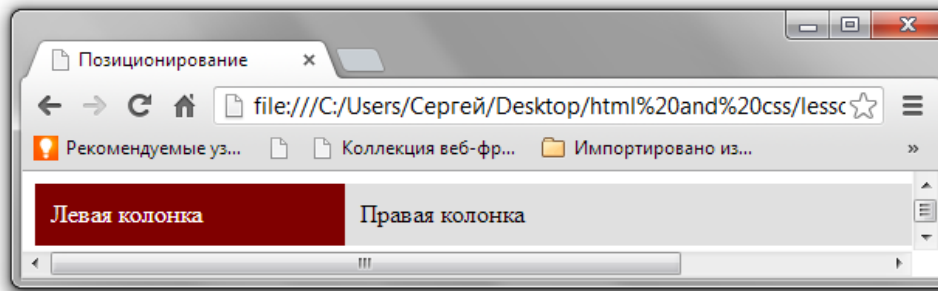
Положение правого слоя `rightcol` зависит от значения стилевого свойства `left`, оно в данном случае должно равняться ширине левого слоя, в этом случае слои будут плотно прилегать между собой. Впрочем, ничего не мешает изменять этот параметр в ту или иную сторону. Тогда между слоями появится разделительная 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>
```

Результат:



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