

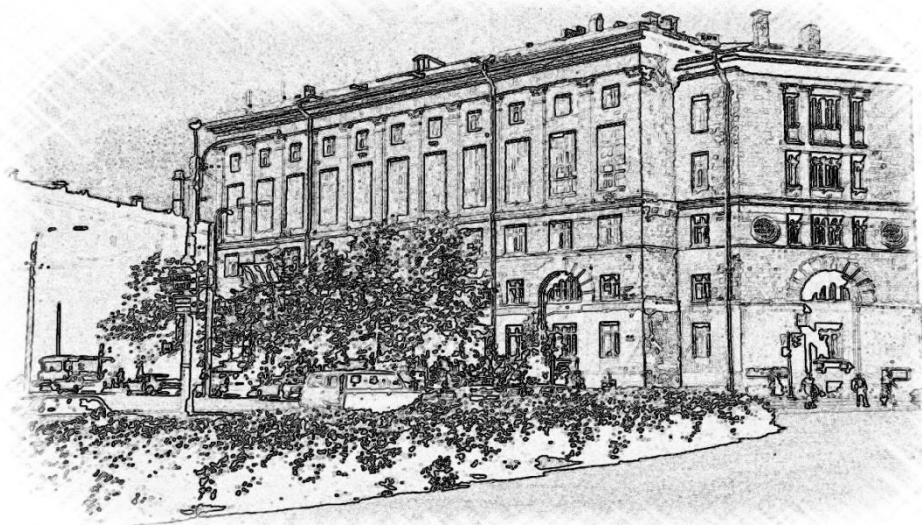
# **МИНОБРНАУКИ РОССИИ**

федеральное государственное автономное образовательное учреждение  
высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»  
(ФГАОУ ВО «СПбПУ»)

Университетский политехнический колледж

## **Практикум по предмету «Технологии публикации цифровой мультимедийной информации»**



Санкт-Петербург

2018

# СОЗДАНИЕ САЙТА

## 1. Исходные данные:

### Требования к созданию сайта:

- 1) Объем страниц не менее 8-ми;
- 2) Тема, содержание и дизайн должны соответствовать друг другу;
- 3) На страницах обязательно использовать структурные теги, заголовок, параграф, ссылки, изображения, аудио и видео в соответствии с логической нагрузкой надписи.  
Например, `<h1> название сайта</h1>`;  
Недопустим «перехлест» тегов.  
Например:  
`<p> <span> <p> <span>`
- 4) Для описания стилей оформления элементов обязательно использование css; Имена классов и идентификаторов – осмысленные.  
Например, header, menu, footer, news.
- 5) Не смешивать HTML и CSS в одном файле;
- 6) Адаптивная верстка сайта, валидный, семантически правильный код, который соответствует стандартам W3C.
- 7) Необходимо выделять комментариями контентную часть, шапку, блоки меню, новостей и т.д. Примерно так:  
`<!--BEGIN FOOTER -->`  
...  
`<!--END FOOTER -->`
- 8) Общая кодировка всех html-файлов UTF-8.
- 9) Кроссбраузерность. Верстка (сайт) должна соответствовать стандартам W3C, нормально работать и обеспечивать одинаково корректное отображение данных в следующих браузерах: Internet Explorer 8+, Mozilla Firefox 35+, Chrome 40+
- 10) Обязательное наличие изображений, аудио и видео-контента;
- 11) Корректно уменьшенные графические изображения на странице.  
При отключении графики страница сохраняет свою функциональность и внешний вид. Использовать атрибут alt тега `<img>`.
- 12) Реализовать выпадающее меню при помощи списков и правил css;
- 13) На одной из страниц создать форму регистрации;
- 14) На одной из страниц создать слайдер;
- 15) Следует создать раздел с контактными данными.

### Требования к содержанию отчета:

Отчет должен быть выполнен в формате word и содержать:

1. Титульный лист
2. Задание на практику
3. Содержание
4. Описание сайта, его темы и целей
5. Описание структуры сайта
6. Дизайн макета главной страницы
7. Скриншоты основных страниц
8. Код html-разметки для всех страниц сайта с комментариями
9. Код файла css с комментариями и описанием стилей
10. Скриншоты валидации html и css
11. Указать домен для просмотра сайта в сети интернет.

### Темы сайта:

- 1) Биография ученого
- 2) Петербургский зоопарк
- 3) География озер Карелии
- 4) Футбольные команды премьер лиги
- 5) Личная страница
- 6) Каталог (автомобили, электроника, ...)
- 7) Учебник (небольшой сайт на выбранную тему с разделами и подразделами)

Дан общий список тем. Возможно расширение данного списка. Определенная тема выбирается самостоятельно и согласуется с преподавателем.

## **2. Инструменты для разработки**

1. Фотошоп/photoshop — графический редактор, используемый веб-дизайнером, поэтому и верстальщик должен иметь данную программу для просмотра и использования частей сайта. Программа платная. Можно использовать различные аналоги. Для данной работы необязательный компонент. Достаточно любого графического редактора для обработки графических файлов перед публикацией.
2. Текстовый редактор (блокнот, Notepad++, SublimeText, ...). Программа бесплатная.
3. Сервер (Денвер, OpenServer, ...) Необязательный компонент для разработки сайта с использованием только HTML/CSS.

### 3. Источники информации

1. <http://htmlbook.ru> – наверное самый/один из самых полезных ресурсов по html и css. Если Вы начинаете, то однозначно необходимо начинать с этого ресурса.
2. [validator.w3.org](http://validator.w3.org) – сервис проверки валидности html кода.
3. [jigsaw.w3.org/css-validator](http://jigsaw.w3.org/css-validator) сервис проверки валидности css кода.
4. <https://htmlacademy.ru/> - Интересные, наглядные и затягивающие курсы, интерактивные интерфейсы, достижения — всё для обучения с удовольствием.
5. <https://html5book.ru> - Сайт для тех, кто изучает веб-технологии и создает сайты
6. [caniuse.com](http://caniuse.com) – для проверки поддержки CSS свойств браузерами.
7. <http://imageslidermaker.com> – сайт с готовыми шаблонами для слайдера

### 4. Порядок выполнения работы

#### 1. Выбор темы. Актуализация темы.

Краткое описание темы будущего сайта и его основных разделов. Формулировка цели создания сайта, например, сайт для описания природного заповедника и т.д. ...

Необходимо на <https://wordstat.yandex.ru/> посмотреть статистику запросов по теме вашего сайта и сделать скрин. Также можно уточнить регион или посмотреть помесечный и недельный срез.

#### 2. Анализ сайтов по аналогичной тематике.

Открыть топ 3 сайта на указанную тему в поисковике, посмотреть и сделать скриншоты главных страниц, добавить в отчет во время оформления.

#### 3. Создание структуры папок и внутренней структуры сайта. Сбор материала

Как известно, при разработке сайта сперва необходимо создать структуру папок сайта. Создайте следующие папки (рис.1.1), где должны храниться файлы и материалы (изображения, видео и т.д.): папку (www) для хранения HTML-документов (index.html, ...) и в созданной папке папку css для хранения каскадной таблицы стилей (styles.css) и папку image для хранения изображений и т.д..



Рис. 1.1 структура папок сайта

Получается следующая структура папок:

В папке **www** содержатся все наши папки с необходимой информацией для сайта, также в папке **www** содержится файл **index.html**.

В папке **audio** аудио файлы и т.д.

Для разработки внутренней структуры сайта сперва надо определить тему сайта и его основные страницы. Это могут быть: главная страница, страница контактов, страницы разделов и подразделов. Следует также определить взаимосвязи между страницами (рис.1.2).

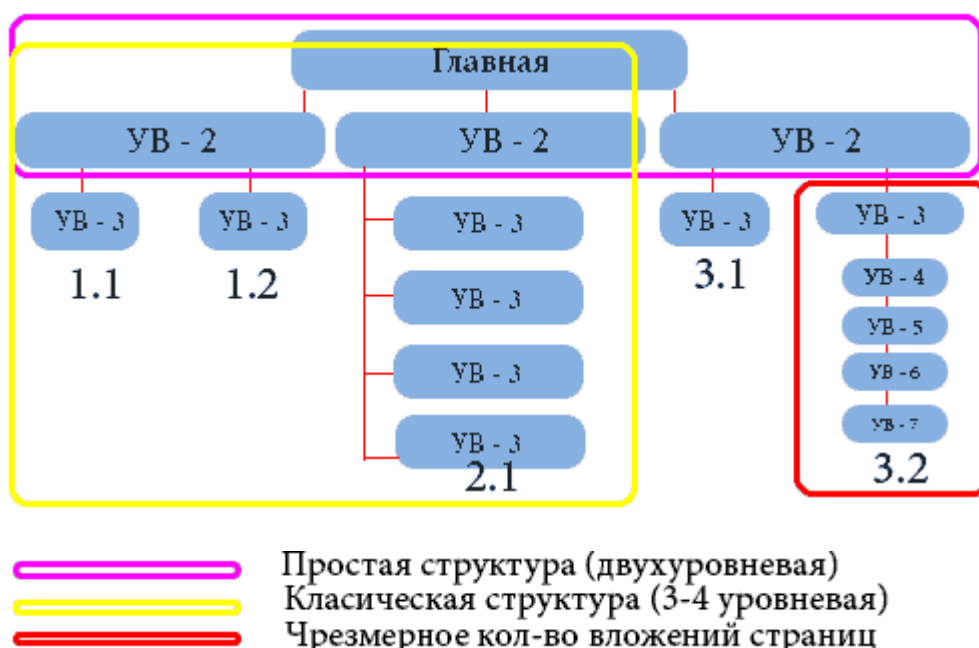


Рис 1.2 внутренняя структура сайта

#### 4. Дизайн макета главной страницы

Для каждой страницы сайта необходимо определить, где должен располагаться основной текст, меню сайта, различные новостные данные, дополнительные и информационные материалы и т.д. Допустим, мы выбрали логическую схему сайта, показанную на рис.1.3. Как показано на рисунке, страница сайта состоит из следующих семи блоков:

- 1 – корпус сайта (site-blok);
- 2 – шапка сайта (header);
- 3 – меню верхнее (menu);
- 4 – блок контента (blok-content);

- 5 – левый блок (blok-left);
- 6 – правый блок (blok-right);
- 7 – подвал сайта(footer);



Рис. 1.3 Макет страницы

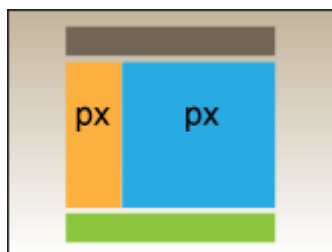
Макет страницы может быть различным. На рисунке 1.4 представлены типовые макеты веб-страниц.

a



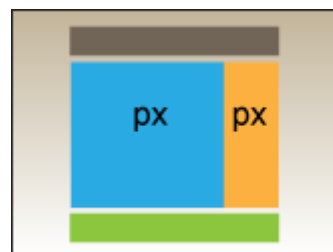
Слой по центру веб-страницы

b



Фиксированная ширина, навигация слева

c

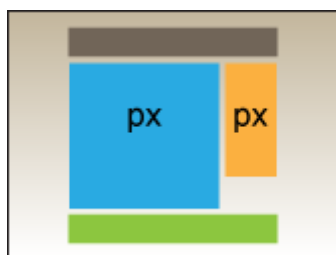


Фиксированная ширина, использование float

d

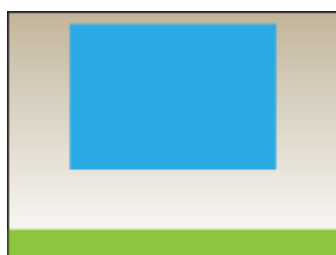
e

f

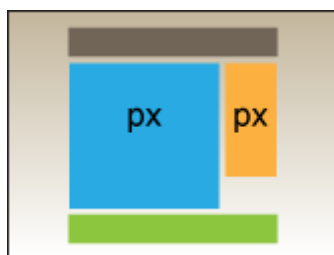


Фиксированная ширина,  
float плюс margin

g

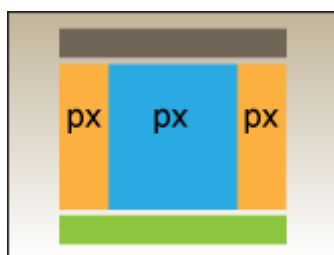


Фиксированный подвал

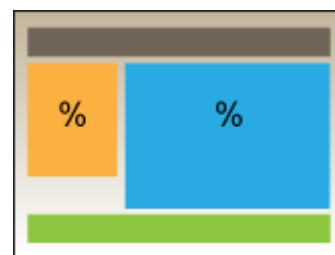


Фиксированная ширина,  
позиционирование

h



Фиксированная ширина,  
три колонки



Резиновая ширина,  
навигация слева

Рис. 1.4 Типовые макеты веб-страниц

## 5. Создание HTML-документа

В папке `www` создайте HTML-документ (`index.html`) выполнив следующие действия:

1. запустить текстовый редактор Блокнот (Стандартные-Блокнот) или HTML-редактор Notepad++;
2. выбрать команду «Файл-Сохранить как» и в появившемся диалоговом окне в поле «Имя файла» введем имя файла с расширением `.html` (рис. 1.6):

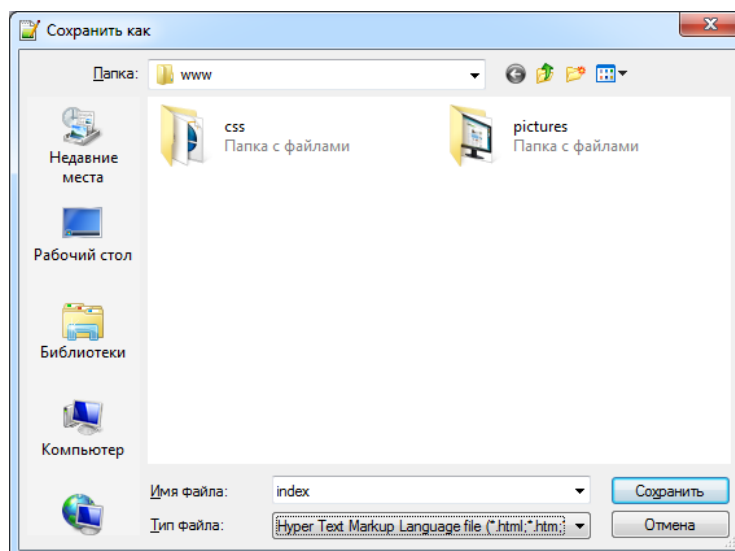
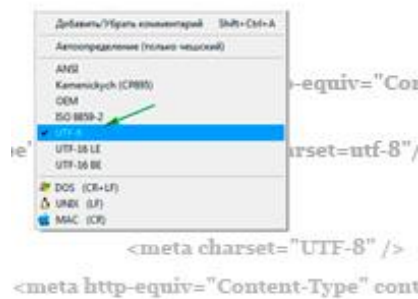


Рис. 1.6.

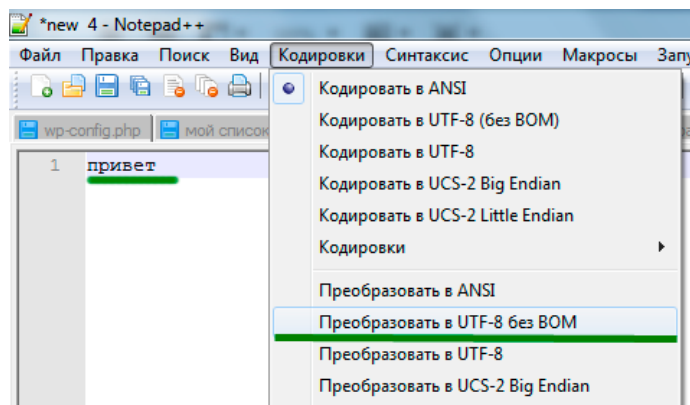
## Кодировка файлов: настройка Notepad++



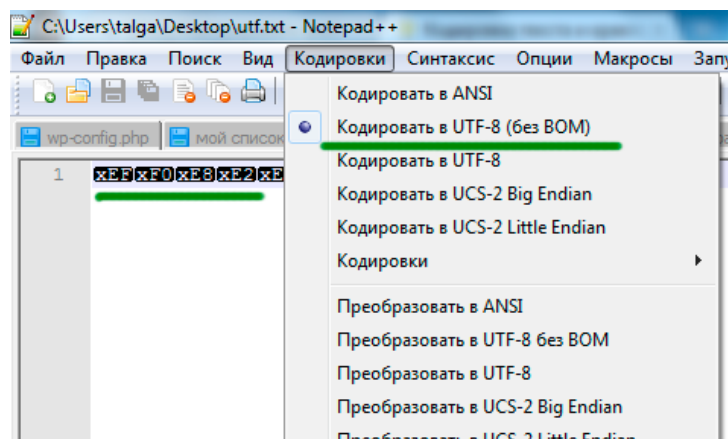
### Кодировка utf-8

Основной проблемой неправильной кодировки это отображение текста. Так же в кодировке window-1251 отсутствуют специальные символы. Все это от того, что в cp1251, например, всего 256 символов (что в принципе достаточно для кириллицы), а в Unicode более 100 000 символов. Поэтому создавать html каркас и вообще сайты лучше с кодировкой utf-8.

Откройте файл и выберите преобразовать в utf-8 без BOM.



Если нажмете кодировать в utf-8 без BOM, то произойдет следующее.





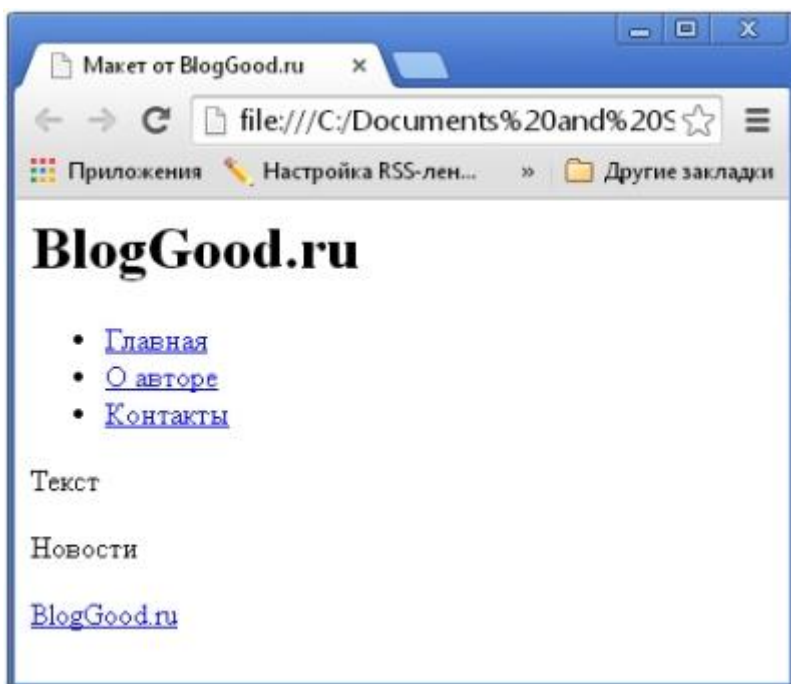
Создайте структуру HTML-документа:

Вот примерная структура HTML страницы:

```
<html>
<head>
<title>Макет от BlogGood.ru</title>
</head>
<body>
<div class="site-blok">
  <div class="header">
    <h1>BlogGood.ru</h1>
    <div class="menu">
      <ul>
        <li><a href="index.html">Главная</a></li>
        <li><a href="index.html">О авторе</a></li>
        <li><a href="index.html">Контакты</a></li>
      </ul>
    </div>
  </div>

  <div class="blok-content">
    <div class="blok-left"><p>Текст</p></div>
    <div class="blok-right"><p>Новости</p></div>
  </div>

  <div class="footer">
    <p><a href="index.html">BlogGood.ru</a></p>
  </div>
</div>
</body>
</html>
```



Аналогично создаем все остальные страницы сайта.

Для просмотра созданного HTML-документа в браузере выполните следующие действия:

1. кликнуть по файлу два раза левой кнопкой мыши, в этом случае веб-страница откроется в том браузере, который выбран по умолчанию;
2. вызвать контекстное меню, выбрать команду «Открыть с помощью...» и выбрать нужный вам браузер.

Примечание: Чтобы отредактировать код HTML-документа, необходимо вызвать контекстное меню и выбрать команду «Открыть с помощью...» и из списка выбрать нужный вам редактор, если в списке нет нужного редактора, выбрать команду «Выбрать программу...», затем из появившегося списка выбрать нужный редактор.

Некоторые редакторы, такие как Notepad++, можно выбрать прямо в контекстном меню, выбрав команду Edit with Notepad++ (Рис. 1.8).

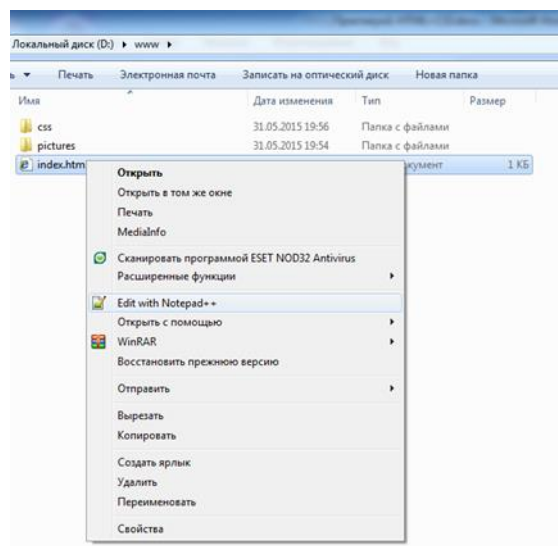


Рис. 1.8.

## 6. Создание каскадной таблицы стилей

В отдельном файле /css/style.css указываем стили для различных элементов сайта. CSS файл(-ы) должен быть разбит на логические блоки.

Комментариями обозначены начало/конец крупных модулей/блоков разметки. CSS файл должен быть разбит с помощью строк с комментариями на блоки по функциональному назначению, например:

```
/* _____ 1. Сброс CSS _____ */
/* _____ 2. Типовые элементы _____ */
/* _____ 2.1. Залоговки _____ */
/* _____ 2.2. Ссылки _____ */
/* _____ 2.3. Элементы форм _____ */
/* _____ 3. HEADER (Шапка сайта) _____ */
/* _____ 4. FOOTER (Подвал ) _____ */
/* _____ 5. SIDEBAR (Справа) _____ */
```

Сброс стилей можно выполнить, подключив CSS файл reset.css следующего содержания:

```
/* http://meyerweb.com/eric/tools/css/reset/
   v2.0 | 20110126
   License: none (public domain)
*/
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
```

```

dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}
body {
    line-height: 1;
}
ol, ul {
    list-style: none;
}
blockquote, q {
    quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
    content: "";
    content: none;
}
table {
    border-collapse: collapse;
    border-spacing: 0;
}

```

Затем следует подключить свой собственный файл стилей styles.css.

Название классов и стилей должны быть логически верными: header, header\_logo, left\_menu, footer, news.

Фактически, в этом случае все имена классов на странице будут уникальными и конфликта стилей не будет.

В папке css создайте CSS документ (styles.css) выполнив следующие действия:

- 1) создать новый документ в редакторе Notepad++;
- 2) выбрать команду «Файл-Сохранить как» и в появившемся диалоговом окне в поле «Имя файла» введем имя файла styles с расширением .css.

Откроем HTML-документ index.html и между тегами <head> ...</head> напишем тег <link> для связывания HTML документа с css файлом (рис. 1.9):



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
5 <link rel="stylesheet" href="css/styles.css" type="text/css">
6 <title>Главная</title>
7 </head>
```

Рис. 1.9.

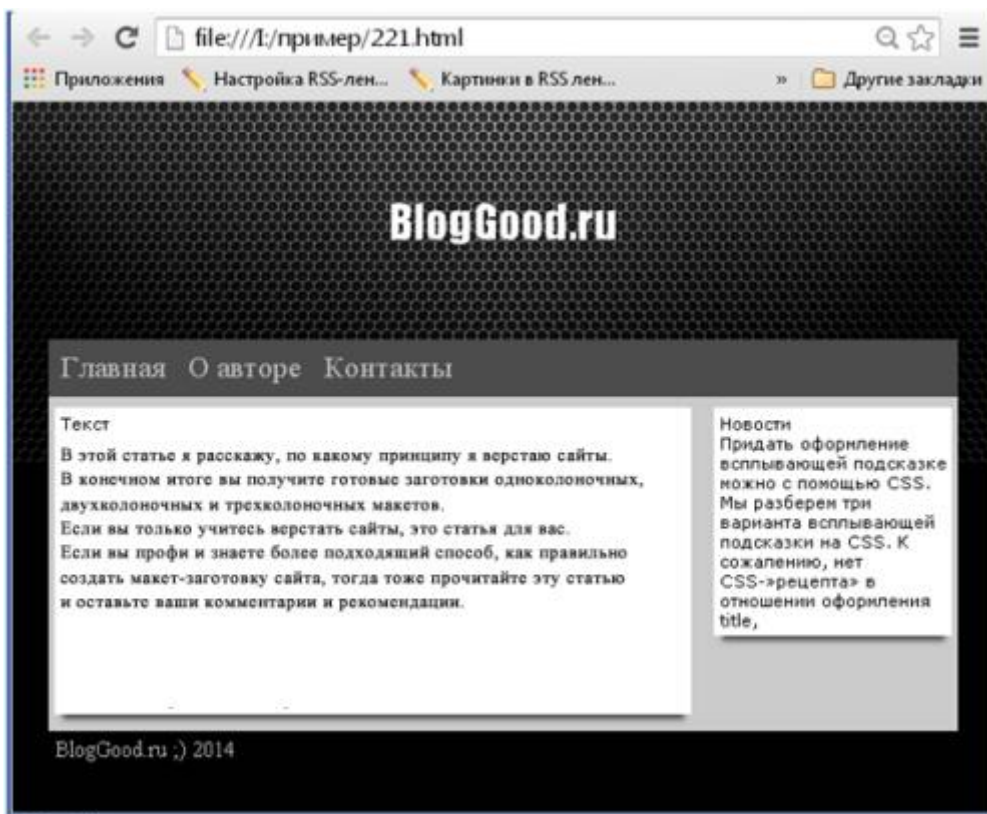
Значение href задаёт путь к CSS-файлу, файл styles.css находится в папке css.

Пример кода файла styles.css:

```
body
{
background:#010101 url(images/fon.png) no-repeat center top;
font:14px Verdana, Arial, Helvetica, sans-serif;
color:#ccc;
}
.site-blok
{
width:800px; /* ширина сайта*/
margin: 0 auto; /* выравниваем сайт по центру*/
}
.header h1
{
margin:80px 0;
text-align:center;
font:40px "Impact", Times, serif;
color:#fff;
}
.menu
{
height:50px;
background:#4c4c4c;
}
.menu ul
{
overflow: hidden;
}
.menu li
{
padding:10px;
float:left; /* граница сайта */
}
.menu li a
{
font:27px "Times New Roman", Times, serif;
color:#bcbcbc;
text-decoration: none;
}
```

```
.blok-content
{
background:#cbcbcb;
overflow: hidden;
padding:10px 5px 15px 5px;
}
.blok-left
{
background:#fff;
width:550px; /* ширина блока левого*/
float:left; /* граница сайта */
padding:5px;
-webkit-box-shadow: 0 10px 6px -6px #444343;
-moz-box-shadow: 0 10px 6px -6px #444343;
box-shadow: 0 10px 6px -6px #44434
}
.blok-right
{
background:#fff;
width:200px; /* ширина блока правого*/
float:right; /* граница сайта */
padding:5px;
-webkit-box-shadow: 0 10px 6px -6px #444343;
-moz-box-shadow: 0 10px 6px -6px #444343;
box-shadow: 0 10px 6px -6px #444343;
}
.footer
{
padding:5px;
}
.footer a
{
font:20px "Times New Roman", Times, serif;
color:#bcbcbc;
text-decoration: none;
}
```

**Результат:**



## 7. Ссылки

1. Ссылки не должны иметь стандартный вид стилей.
2. Все визуальные украшения (цвет, рамки, тени, эффекты и т.д.) должны быть сделаны с помощью стилей, а не быть включенными в ссылку.

**Пример:** `<a href="index.html">Главная</a>`

## 8. Изображения

1. Нужно использовать png для схем и скриншотов, jpg для фотографий.
2. У картинок должен быть указан alt и title.
3. Изображения должны масштабироваться в зависимости от размера окна (`max-width: 100%; height: auto;`).
4. Изображения должны быть оптимизированы.
5. Все визуальные украшения (рамки, тени) должны быть сделаны с помощью стилей, а не быть включенными в картинку.
6. Стилиевые (стрелки, иконки, фоны) и содержательные картинки (скриншоты, фотографии) должны лежать в разных папках.

**Пример:** ``

## 9. Добавление аудио и видео-контента на сайт.

Чтобы вставить аудио-файл, достаточно написать код в таком виде :

`<audio id="myAudio" controls loop>`

```
<source src="elvis.mp3" />
<source src="elvis.ogg" />
</audio>
```

Обратите внимание на две вещи:

1. Наличие атрибутов без значений – это так называемые бинарные атрибуты, наличие которых эквивалентно значению true или тому, которое определяется стандартом. То есть controls равносильно тому, что вы бы написали controls = “controls”.
2. В данном случае указано два источника для воспроизведения, чтобы браузер мог выбрать, какой из них он может проиграть в зависимости от того, какие кодеки он поддерживает (к этому вопросу мы вернемся в следующем разделе). Если источник только один, то достаточно указать ссылку на файл в родительском теге audio через атрибут src.

Вставка видео производится абсолютно аналогично, но с некоторыми дополнительными возможностями (размер и постер):

```
<video id="myVideo" width="640" height="480" poster="images/elvis.jpg" >
  <source src="http://media.w3.org/2010/05/sintel/trailer.mp4"
    type='video/mp4; codecs="avc1, mp4a"'>
  <source src="http://media.w3.org/2010/05/sintel/trailer.ogv"
    type='video/ogg; codecs="theora, vorbis"'>
  <p>Your user agent does not support the HTML5 Video element.</p>
</video>
```

Постер – это картинка, которая показывается до начала воспроизведения.

Когда вы вставите аудио или видео на страницу, браузер предложит вам готовые средства для управления воспроизведением. Важный нюанс, который в отдельных сценариях может иметь значение – эти элементы воспроизведения видео или аудио – разные в разных браузерах и не подлежат стилизации через CSS или какие-либо другие механизмы.

В большинстве случаев это не критично, но если вы хотите, чтобы контролы выглядели одинаково или выглядели совсем по-другому, все в ваших силах. Через JavaScript можно управлять воспроизведением, громкостью, узнавать текущую отсечку во времени проигрывания и длительность всего произведения, а также ряд других параметров.

## 10. Слайдер.

Визуально слайдер для сайта представляет собой **блок на странице**, в пределах которого с определенной периодичностью демонстрируются анонсы статей или новостей либо изображения.

Создание слайдера.



Откройте HTML-файл и вставьте этот код в то место, где вы хотите, чтобы отображался слайдер (но обязательно между тегами <body></body>).

```
1
2      <!--слайдер на блоге BlogGood.ru -->
3      <div class="slides">
4          <ul>
5              <li>
6                  <div>Текст для слайдера #1</div>
7              </li>
8              <li>
9                  <div>Текст для слайдера #2</div>
10             </li>
11             <li>
12                 <div>Текст для слайдера #3</div>
13             </li>
14             <li>
15                 <div>Текст для слайдера #4</div>
16             </li>
17             <!-- сюда добавляем по необходимости еще слайды -->
18         </ul>
19     </div>
```

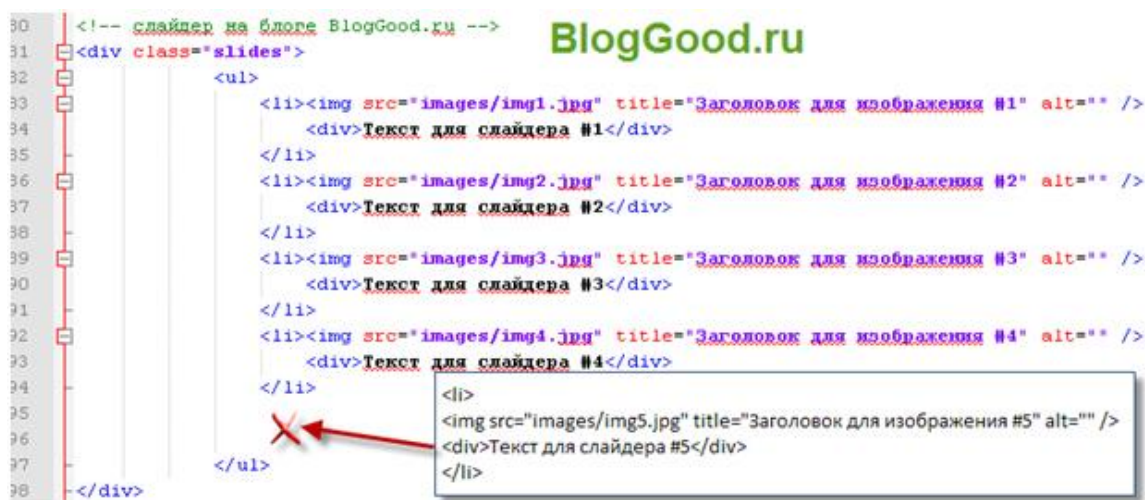
Как видите, в коде четыре слайда. Туда входит картинка (img1.jpg, img2.jpg, img3.jpg, img4.jpg), описание к картинке (title) и текст для картинки.

```
1      <li>
2          
3          <div>Текст для слайдера #1</div>
4      </li>
```

Если вам нужно добавить больше слайдов, достаточно добавить вот эту строчку перед </ul>:

```
1      <li>
2          
3          <div>Текст для слайдера #5</div>
4      </li>
```

Пример с картинкой:



Теперь создайте CSS файл и вставьте следующий код:

```
/* ползунок */
.slides {
height:300px;
margin:50px auto;
overflow:hidden;
position:relative;
width:900px;
}
.slides ul {
list-style:none;
position:relative;
}

/* keyframes #anim_slides */
@-webkit-keyframes anim_slides {
0% {
opacity:0;
}
6% {
opacity:1;
}
24% {
opacity:1;
}
30% {
opacity:0;
}
100% {
opacity:0;
}
}
@-moz-keyframes anim_slides {
0% {
opacity:0;
}
6% {
opacity:1;
}
24% {
opacity:1;
}
30% {
opacity:0;
}
100% {
opacity:0;
}
}

.slides ul li {
opacity:0;
position:absolute;
top:0;

/* css3 анимация */
-webkit-animation-name: anim_slides;
-webkit-animation-duration: 24.0s;
-webkit-animation-timing-function: linear;
-webkit-animation-iteration-count: infinite;
-webkit-animation-direction: normal;
-webkit-animation-delay: 0;
```

```

-webkit-animation-play-state: running;
-webkit-animation-fill-mode: forwards;

-moz-animation-name: anim_slides;
-moz-animation-duration: 24.0s;
-moz-animation-timing-function: linear;
-moz-animation-iteration-count: infinite;
-moz-animation-direction: normal;
-moz-animation-delay: 0;
-moz-animation-play-state: running;
-moz-animation-fill-mode: forwards;
}

/* css3 delays */
.slides ul li:nth-child(2), .slides ul li:nth-child(2) div {
-webkit-animation-delay: 6.0s;
-moz-animation-delay: 6.0s;
}
.slides ul li:nth-child(3), .slides ul li:nth-child(3) div {
-webkit-animation-delay: 12.0s;
-moz-animation-delay: 12.0s;
}
.slides ul li:nth-child(4), .slides ul li:nth-child(4) div {
-webkit-animation-delay: 18.0s;
-moz-animation-delay: 18.0s;
}
.slides ul li img {
display:block;
}

/* keyframes #anim_titles */
@-webkit-keyframes anim_titles {
0% {
left:100%;
opacity:0;
}
5% {
left:10%;
opacity:1;
}
20% {
left:10%;
opacity:1;
}
25% {
left:100%;
opacity:0;
}
100% {
left:100%;
opacity:0;
}
}
@-moz-keyframes anim_titles {
0% {
left:100%;
opacity:0;
}
5% {
left:10%;
opacity:1;
}
20% {
left:10%;

```

```

opacity:1;
}
25% {
left:100%;
opacity:0;
}
100% {
left:100%;
opacity:0;
}
}

.slides ul li div {
background-color:#000000;
border-radius:10px 10px 10px 10px;
box-shadow:0 0 5px #FFFFFF inset;
color:#FFFFFF;
font-size:26px;
left:10%;
margin:0 auto;
padding:20px;
position:absolute;
top:50%;
width:200px;

/* css3 анимация */
-webkit-animation-name: anim_titles;
-webkit-animation-duration: 24.0s;
-webkit-animation-timing-function: linear;
-webkit-animation-iteration-count: infinite;
-webkit-animation-direction: normal;
-webkit-animation-delay: 0;
-webkit-animation-play-state: running;
-webkit-animation-fill-mode: forwards;

-moz-animation-name: anim_titles;
-moz-animation-duration: 24.0s;
-moz-animation-timing-function: linear;
-moz-animation-iteration-count: infinite;
-moz-animation-direction: normal;
-moz-animation-delay: 0;
-moz-animation-play-state: running;
-moz-animation-fill-mode: forwards;
}

```

**посмотреть демонстрацию этого слайдера:**

<http://bloggood.ru/wp-content/lessons/css/fx-css-2/1.html>

Также вы можете сделать любой другой слайдер и воспользоваться любыми источниками по созданию слайдера, например:

<http://imageslidermaker.com/>

## 11. Форма.

**Форма** — это инструмент, с помощью которого HTML-документ может отправить информацию в заранее определенную точку внешнего мира. Формы применяются для опроса посетителей, покупки чего-либо (заказа товара(ов)), отправки электронной почты.

Принцип работы форм следующий: посетитель, зашедший к вам на сайт заполняет определенную форму (вносит туда нужные данные), а после нажатия определенной кнопки форма берет данные из заполненных полей и отправляет их в назначенное место.

Формы размещаются между тэгами **<FORM>** **</FORM>**. HTML-документ может содержать в себе несколько форм, но они не должны находиться одна внутри другой. Тэг **<FORM>** может содержать следующие атрибуты:

<b>ACTION</b>	<b>Обязательный атрибут.</b> Определяет, где находится обработчик формы.
<b>METHOD</b>	Определяет, каким образом данные из формы будут переданы обработчику. Допустимые значения: <b>METHOD=POST</b> и <b>METHOD=GET</b> . Если значение атрибута не установлено, по умолчанию предполагается <b>METHOD=GET</b> .
<b>ENCTYPE</b>	Определяет, каким образом данные из формы будут закодированы для передачи обработчику.

**Обратите внимание на первую строку в таблице выше.** В правом столбце Вы прочитали такую фразу: ... **обработчик формы**. Это значит, что кроме файла с формой (куда вносятся данные) **существует еще один файл (или скрипт), который обрабатывает полученные из формы данные, а уже затем отправляет дальше.** Обычно это происходит скрыто и посетитель сайта даже не догадывается об этом. Но про Обработчик мы поговорим позже.

Для внесения информации пользователем в форму используется элемент **<INPUT>** Это и есть поля, в которые пользователь вводит информацию. Каждый элемент **<INPUT>** включает атрибут **NAME=имя**, определяющий имя данного поля (идентификатор поля). В таблице ниже представлены основные типы, применяемых элементов **<INPUT>**:

<b>TYPE=text</b>	<p>Определяет окно для ввода строки текста. Может содержать дополнительные атрибуты SIZE=число (ширина окна ввода в символах) и MAXLENGTH=число (максимально допустимая длина вводимой строки в символах):</p> <p><b>&lt;INPUT TYPE=text SIZE=20 NAME=User VALUE="Сайт LuksWeb.ru"&gt;</b></p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 10px auto;">Сайт LuksWeb.ru</div> <p>Простая форма с полем шириной 20 символов для ввода текста. По умолчанию в окне находится текст <b>Сайт LuksWeb.ru</b>, который пользователь может изменить.</p>
<b>TYPE=password</b>	<p>Определяет окно для ввода пароля. Абсолютно аналогичен типу <b>text</b>, только вместо символов вводимого текста показывает на экране звездочки (*):</p> <p><b>&lt;INPUT TYPE=password NAME=PW SIZE=20 MAXLENGTH=10&gt;</b></p>

	<div></div>
<b>TYPE=radio</b>	<p>Определяет радиокнопку. Может содержать дополнительный атрибут <b>CHECKED</b> (показывает, что кнопка отмечена). В группе радиокнопок с одинаковыми именами может быть только одна помеченная радиокнопка:</p> <pre>&lt;INPUT TYPE=radio NAME=Question VALUE="Yes" CHECKED&gt; Да &lt;INPUT TYPE=radio NAME=Question VALUE="No"&gt; Нет &lt;INPUT TYPE=radio NAME=Question VALUE="Possible"&gt; Возможно</pre> <p> <input checked="" type="radio"/> Да  <input type="radio"/> Нет  <input type="radio"/> Возможно </p> <p>Определяет группу из трех радиокнопок, подписанных <b>Yes</b>, <b>No</b> и <b>Possible</b>. Первоначально помечена первая из кнопок. Если пользователь не отметит другую кнопку, обработчику будет передана переменная <b>Question</b> со значением <b>Yes</b>. Если пользователь отметит другую кнопку, обработчику будет передана переменная <b>Question</b> со значением <b>No</b> или <b>Possible</b>. <b>Одновременно можно выбрать только одно значение.</b></p>
<b>TYPE=checkbox</b>	<p>Определяет квадрат, в котором можно сделать пометку. Может содержать дополнительный атрибут <b>CHECKED</b> (показывает, что квадрат помечен). В отличие от радиокнопок, в группе квадратов с одинаковыми именами может быть несколько помеченных квадратов:</p> <pre>&lt;INPUT TYPE=checkbox NAME=Comp VALUE="CPU"&gt; Процессоры &lt;INPUT TYPE=checkbox NAME=Comp VALUE="Video" CHECKED&gt; Видеоадаптеры &lt;INPUT TYPE=checkbox NAME=Comp VALUE="Scan"&gt; Сканеры &lt;INPUT TYPE=checkbox NAME=Comp VALUE="Modem" CHECKED&gt; Модемы</pre> <p> <input type="checkbox"/> Процессоры  <input checked="" type="checkbox"/> Видеоадаптеры  <input type="checkbox"/> Сканеры  <input checked="" type="checkbox"/> Модемы </p> <p>Определяет группу из четырех квадратов. Первоначально помечены второй и четвертый квадраты. Если пользователь не произведет изменений, обработчику будут переданы две переменные: <b>Comp=Video</b> и <b>Comp=Modem</b>. <b>Одновременно</b></p>

	<b>можно выбрать любое количество значений.</b>
<b>TYPE=hidden</b>	<p>Определяет скрытый элемент данных, который не виден пользователю при заполнении формы и передается обработчику без изменений. Такой элемент иногда полезно иметь в форме, которая время от времени подвергается переработке, чтобы обработчик мог знать, с какой версией формы он имеет дело.</p> <p><b>&lt;INPUT TYPE=hidden NAME=version VALUE="1.1"&gt;</b></p> <p>Определяет скрытую переменную version, которая передается обработчику со значением 1.1.</p>
<b>TYPE=submit</b>	<p>Определяет кнопку, при нажатии на которую запускается процесс передачи данных из формы обработчику:</p> <p><b>&lt;INPUT TYPE=submit VALUE="Отправить"&gt;</b></p> 
<b>TYPE=reset</b>	<p>Определяет кнопку, при нажатии на которую очищаются поля формы. Поскольку при использовании этой кнопки данные обработчику не передаются, кнопка типа reset может и не иметь атрибута name:</p> <p><b>&lt;INPUT TYPE=reset VALUE=" Сброс "&gt;</b></p> 

Формы могут содержать поля для ввода большого текста **<TEXTAREA>**:

**<TEXTAREA NAME=address ROWS=5 COLS=50> Наберите здесь своё сообщение</TEXTAREA>**

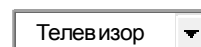


Атрибут **NAME** определяет имя, под которым содержимое окна будет передано обработчику. Атрибут **ROWS** устанавливает высоту окна в строках. Атрибут **COLS** устанавливает ширину окна в символах. Текст, размещенный между тэгами **<TEXTAREA>** **</TEXTAREA>**, представляет собой содержимое окна по умолчанию. Пользователь может его отредактировать или просто стереть.

Кроме всего этого все формы могут содержать меню выбора или как иногда говорят **"Выпадающий список"**, которое начинается открывающимся

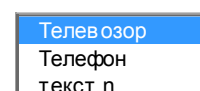
тэгом **<SELECT>** (содержит обязательный атрибут **NAME**, определяющий имя меню) и завершается закрывающимся **</SELECT>**. Между ними находятся тэги **<OPTION>**, определяющие элемент меню. Обязательный атрибут **VALUE** устанавливает значение, которое будет передано обработчику, если выбран этот элемент меню. Тэг **<OPTION>** может включать атрибут **selected**, показывающий, что данный элемент выбран/отмечен по умолчанию.

```
<SELECT NAME="имя">
<OPTION VALUE="option_1"
selected>Телевизор
<OPTION VALUE="option_2">Телефон
<OPTION VALUE="option_n">текст n
</SELECT>
```



Тэг **<SELECT>** может также содержать атрибут **MULTIPLE**, присутствие которого показывает, что из меню можно выбрать несколько элементов. Большинство Браузеров показывают меню **<SELECT MULTIPLE>** в виде окна, в котором находятся элементы меню. Высоту окна в строках можно задать атрибутом **SIZE=число**.

```
<SELECT MULTIPLE SIZE=3 NAME="имя">
<OPTION VALUE="option_1"
selected>Телевизор
<OPTION VALUE="option_2">Телефон
<OPTION VALUE="option_n">текст n
</SELECT>
```



Пример:

Откройте HTML-файл и вставьте этот код в то место, где вы хотите, чтобы отображалась форма(но обязательно между тэгами **<body></body>**).

```
<form class="transparent">
  <div class="form-inner">
    <h3>Регистрация</h3>
    <label for="username">Имя пользователя</label>
    <input type="text" id="username">
    <label for="password">Пароль</label>
    <input id="password" type="password">
    <input type="checkbox" id="custom-checkbox">
    <label for="custom-checkbox">Запомнить меня</label>
    <input type="submit" value="Отправить">
  </div>
</form>
```

Теперь создайте CSS файл и вставьте следующий код:

```
* {box-sizing: border-box;}
```



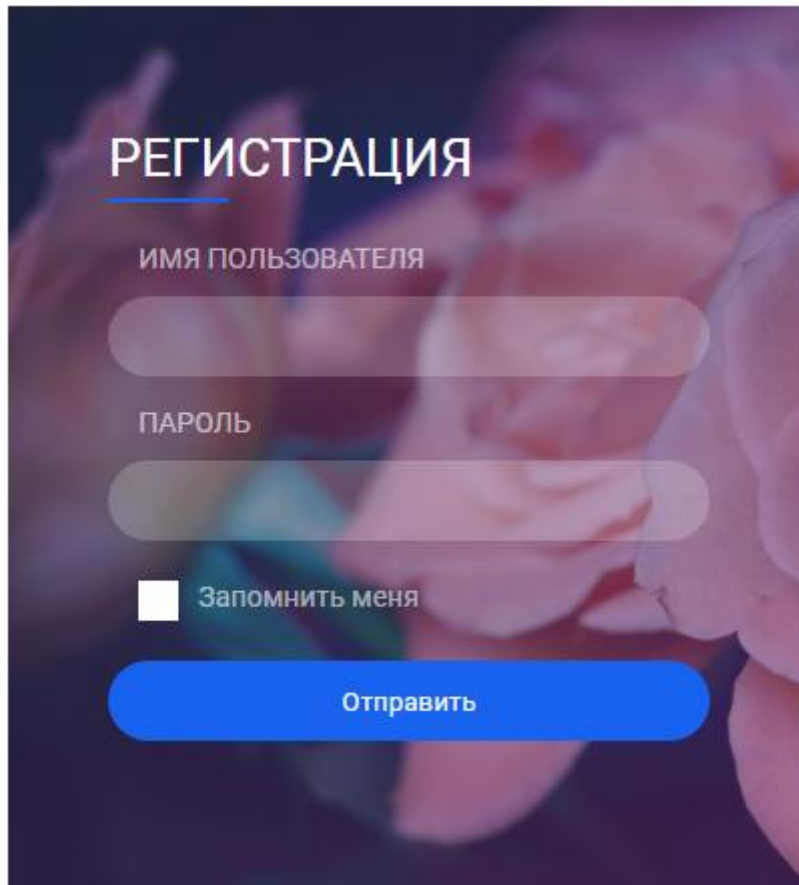
```
.transparent {
  position: relative;
  max-width: 400px;
  padding: 60px 50px;
  margin: 50px auto 0;
  background-image: url(https://html5book.ru/wp-content/uploads/2017/01/photo-roses.jpg);
  background-size: cover;
}
.transparent:before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background: linear-gradient(to right bottom, rgba(43, 44, 78, .5), rgba(104, 22, 96, .5));
}
.form-inner {position: relative;}
.form-inner h3 {
  position: relative;
  margin-top: 0;
  color: white;
  font-family: 'Roboto', sans-serif;
  font-weight: 300;
  font-size: 26px;
  text-transform: uppercase;
}
.form-inner h3:after {
  content: "";
  position: absolute;
  left: 0;
  bottom: -6px;
  height: 2px;
  width: 60px;
  background: #1762EE;
}
.form-inner label {
  display: block;
  padding-left: 15px;
  font-family: 'Roboto', sans-serif;
  color: rgba(255,255,255,.6);
  text-transform: uppercase;
  font-size: 14px;
}
.form-inner input {
  display: block;
  width: 100%;
  padding: 0 15px;
  margin: 10px 0 15px;
  border-width: 0;
```

```

    line-height: 40px;
    border-radius: 20px;
    color: white;
    background: rgba(255,255,255,.2);
    font-family: 'Roboto', sans-serif;
}
.form-inner input[type="checkbox"] {
    position: absolute;
    opacity: 0;
}
#custom-checkbox + label {
    position: relative;
    margin: 20px 0;
    text-transform: none;
    cursor: pointer;
}
#custom-checkbox + label:before {
    content: "";
    display: inline-block;
    width: 20px;
    height: 20px;
    margin-right: 10px;
    vertical-align: text-top;
    background: white;
}
#custom-checkbox:checked + label:before {background: #1762EE;}
#custom-checkbox:checked + label:after {
    content: "";
    position: absolute;
    width: 2px;
    height: 2px;
    left: 20px;
    top: 9px;
    background: white;
    box-shadow: 2px 0 0 white, 4px 0 0 white, 4px -2px 0 white, 4px -4px 0
white, 4px -6px 0 white, 4px -8px 0 white;
    transform: rotate(45deg);
}
.form-inner input[type="submit"] {background: #1762EE;}

```

Результат:



РЕГИСТРАЦИЯ

ИМЯ ПОЛЬЗОВАТЕЛЯ

ПАРОЛЬ

☐ Запомнить меня

Отправить

## 12. Таблица. Разметка HTML

```
<table>
<tr>
  <th colspan="2">Модель</th>
  <th>Цена</th>
  <th>Количество</th>
  <th>Итого</th>
</tr>
<tr>
  <td></td>
  <td>Платье с цветочным принтом</td>
  <td>2500</td>
  <td>1</td>
  <td>2500</td>
</tr>
<tr>
  <td></td>
  <td>Платье с боковыми вставками</td>
  <td>3000</td>
  <td>1</td>
```

```
<td>3000</td>
</tr>
</table>
```

## Стили CSS

```
table {
border-spacing: 0 10px;
font-family: 'Open Sans', sans-serif;
font-weight: bold;
}
th {
padding: 10px 20px;
background: #56433D;
color: #F9C941;
border-right: 2px solid;
font-size: 0.9em;
}
th:first-child {
text-align: left;
}
th:last-child {
border-right: none;
}
td {
vertical-align: middle;
padding: 10px;
font-size: 14px;
text-align: center;
border-top: 2px solid #56433D;
border-bottom: 2px solid #56433D;
border-right: 2px solid #56433D;
}
td:first-child {
border-left: 2px solid #56433D;
border-right: none;
}
td:nth-child(2){
text-align: left;
}
```

Результат:

Модель	Цена	Количество	Итого
 Платье с цветочным принтом	2500	1	2500
 Платье с боковыми вставками	3000	1	3000

### 13. Адаптивная верстка с использованием CSS.

Адаптивная вёрстка сайта позволяет веб-страницам автоматически подстраиваться под экраны планшетов и смартфонов. Мобильный интернет-трафик растёт с каждым годом, и чтобы эффективно обрабатывать этот трафик, нужно предлагать пользователям адаптивные сайты с удобным интерфейсом.

Поисковые системы используют ряд критериев для оценки адаптивности сайта при просмотре на мобильных устройствах. Google старается упростить пользование Интернетом для владельцев смартфонов и планшетов, отмечая в мобильной выдаче адаптированные под мобильные устройства сайты специальной пометкой **mobile-friendly**. В Яндексе также работает алгоритм, который отдаёт предпочтение сайтам с мобильной/адаптивной версией для пользователей в мобильном поиске.

Проверить отображение страницы на мобильных устройствах можно на сервисах [Яндекс.Вебмастер](#) и [Google Developers](#).

Адаптивная вёрстка предполагает отсутствие горизонтальной полосы прокрутки и масштабируемых областей при просмотре на любом устройстве, читабельный текст и большие области для кликабельных элементов. С помощью медиазапросов можно управлять компоновкой и расположением блоков на странице, перестраивая шаблон таким образом, чтобы он адаптировался под разные размеры экранов устройств.

Более подробную информацию об адаптивной верстке можно прочитать здесь:

<https://html5book.ru/otzyvchivyyj-dizayn-saita>

Пример:

#### 1. Метатеги и раздел `<head>`

1) Добавим в раздел `<head>` необходимые файлы — ссылку на используемые шрифты, библиотеку jQuery, а также плагин prefixfree (чтобы не писать для свойств браузерные префиксы):

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
```

```

<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Адаптивная вёрстка сайта</title>
<link rel="stylesheet" type="text/css"
href="https://fonts.googleapis.com/css?family=Open+Sans:400,400italic,600
,600italic,700,700italic|Playfair+Display:400,700&subset=latin,cyrillic">
<link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.4.0/css/font-
awesome.css">
<link rel="stylesheet" type="text/css" href="style.css">
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.2.2/jquery.min.js"><
/script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/prefixfree/1.0.7/prefixfree.m
in.js"></script>
</head>
<body>

```

## 2. Шапка страницы

В шапке страницы `<header>` поместим следующие элементы-контейнеры: логотип `<a class="logo">`;  
 кнопку для показа/скрытия главного меню `<div class="nav-toggle">`;  
 главное меню `<ul id="menu">`;  
 форму поиска по сайту `<form id="searchform">`.

```

<header>
  <nav class="container">
    <a class="logo" href="">
      <span>L</span>
      <span>O</span>
      <span>G</span>
      <span>O</span>
    </a>
    <div class="nav-toggle"><span></span></div>
    <form action="" method="get" id="searchform">
      <input type="text" placeholder="Искать на сайте...">
      <button type="submit"><i class="fa fa-search"></i></button>
    </form>
    <ul id="menu">
      <li><a href="">Блог</a></li>
      <li><a href="">Портфолио</a></li>
      <li><a href="">О6 авторе</a></li>
    </ul>
  </nav>
</header>

```

## 3. Блок с кратким содержанием статьи

Анонс статей обернём элементом `<article id="post-1" class="post">`:

```

<div class="container">

```

```

<div class="posts-list">
  <article id="post-1" class="post">
    <div class="post-image"><a href=""></a></div>
    <div class="post-content">
      <div class="category"><a href="">Дизайн</a></div>
      <h2 class="post-title">Весна</h2>
      <p>Очень богат русский язык словами, относящимися к временам года
и к природным явлениям, с ними связанным.</p>
      <div class="post-footer">
        <a class="more-link" href="">Продолжить чтение</a>
        <div class="post-social">
          <a href="" target="_blank"><i class="fa fa-facebook"></i></a>
          <a href="" target="_blank"><i class="fa fa-twitter"></i></a>
          <a href="" target="_blank"><i class="fa fa-
pinterest"></i></a>
        </div>
      </div>
    </div>
  </article>
  <article id="post-2" class="post">
    ...
  </article>
</div> <!-- конец div class="posts-list"-->

```

#### 4. Боковая колонка

В боковую колонку `<aside>` добавим список категорий, последние записи и форму подписки на рассылку:

```

<aside>
  <div class="widget">
    <h3 class="widget-title">Категории</h3>
    <ul class="widget-category-list">
      <li><a href="">Дизайн</a> (2)</li>
      <li><a href="">Вёрстка</a> (5)</li>
      <li><a href="">Видео</a> (1)</li>
    </ul>
  </div>
  <div class="widget">
    <h3 class="widget-title">Последние записи</h3>
    <ul class="widget-posts-list">
      <li>
        <div class="post-image-small">
          <a href=""></a>
        </div>
        <h4 class="widget-post-title">Весна</h4>
      </li>
      <li>
        <div class="post-image-small">

```

```

        <a href=""></a>
    </div>
    <h4 class="widget-post-title">Лето</h4>
</li>
<li>
    <div class="post-image-small">
        <a href=""></a>
    </div>
    <h4 class="widget-post-title">Осень</h4>
</li>
</ul>
</div>
<div class="widget">
    <h3 class="widget-title">Подписка на рассылку</h3>
    <form action="" method="post" id="subscribe">
        <input type="email" name="email" placeholder="Ваш email" required>
        <button type="submit"><i class="fa fa-paper-plane-o"></i></button>
    </form>
</div>
</aside>
</div> <!-- конец div class="container"-->

```

## 5. Нижний колонтитул

В подвале страницы разместим информацию о копирайте, кнопки социальных сетей и ссылку на электронную почту:

```

<footer>
    <div class="container">
        <div class="footer-col"><span>Мой блог © 2016</span></div>
        <div class="footer-col">
            <div class="social-bar-wrap">
                <a title="Facebook" href="" target="_blank"><i class="fa fa-
facebook"></i></a>
                <a title="Twitter" href="" target="_blank"><i class="fa fa-
twitter"></i></a>
                <a title="Pinterest" href="" target="_blank"><i class="fa fa-
pinterest"></i></a>
                <a title="Instagram" href="" target="_blank"><i class="fa fa-
instagram"></i></a>
            </div>
        </div>
        <div class="footer-col">
            <a href="mailto:admin@yoursite.ru">Написать письмо</a>
        </div>
    </div>
</footer>
<script>
$('.nav-toggle').on('click', function(){
$('#menu').toggleClass('active');
});

```



```
</script>
</body>
</html>
```

## 6. Общие CSS-стили

Общие стили, сброс стилей браузера по умолчанию:

```
*, *:after, *:before {
box-sizing: border-box;
padding: 0;
margin: 0;
transition: .5s ease-in-out; /* добавим плавность переходов для всех
элементов страницы*/
}
ul {
list-style: none;
}
a {
text-decoration: none;
outline: none;
}
img {
display: block;
width: 100%;
}
h1, h2, h3, h4, h5, h6 {
font-family: 'Playfair Display';
font-weight: normal;
letter-spacing: 1px;
}
body {
font-family: 'Open Sans', arial, sans-serif;
font-size: 14px;
line-height: 1;
color: #373737;
background: #f7f7f7;
}
/* добавим очистку потока для всех контейнеров, внутри которых задано
обтекание дочерних элементов */
header:after, .container:after, footer:after, .widget-posts-list
li:after, #subscribe:after {
content: "";
display: table;
clear: both;
}
/* стилевой класс, который управляет шириной контейнера сетки*/
.container {
margin: 0 auto;
width: 100%;
max-width: 960px;
padding: 0 15px;
```

```
}
```

## 7. Стили для шапки и её содержимого

```
header {  
width: 100%;  
background: white;  
box-shadow: 3px 3px 1px rgba(0,0,0,.05);  
padding: 15px 0;  
margin-bottom: 30px;  
position: relative;  
}  
nav {  
width: 100%;  
}  
/* логотип */  
.logo {  
display: block;  
float: left;  
}  
.logo span {  
color: white;  
display: inline-block;  
width: 30px;  
height: 30px;  
line-height: 30px;  
border-radius: 50%;  
margin: 5px 0;  
text-align: center;  
text-shadow: 2px 2px 1px rgba(0,0,0,.4);  
}  
.logo span:nth-child(odd) {  
background: #EF5A42;  
}  
.logo span:nth-child(even) {  
background: #F8B763;  
}  
/* меню */  
#menu {  
float: right;  
}  
#menu li {  
display: inline-block;  
margin-right: 30px;  
}  
#menu a {  
color: #111;  
text-transform: uppercase;  
letter-spacing: 1px;  
font-weight: 600;  
display: block;
```

```
line-height: 40px;
}
#menu a:hover {
color: #EF5A42;
}
#menu li:last-child {
margin-right: 0;
}
/* форма поиска */
#searchform {
float: right;
margin-left: 46px;
display: inline-block;
position: relative;
}
#searchform input {
width: 170px;
float: left;
border: none;
padding-left: 10px;
height: 40px;
overflow: hidden;
outline: none;
color: #9E9C9C;
font-style: italic;
}
#searchform button {
background: transparent;
height: 40px;
border: none;
position: absolute;
right: 10px;
color: #EF5A42;
cursor: pointer;
font-size: 18px;
}
#searchform input:focus {
outline: 2px solid #EBEBE3;
}
/* кнопка переключения меню, появляющаяся при ширине 768px */
.nav-toggle {
display: none;
position: relative;
float: right;
width: 40px;
height: 40px;
margin-left: 20px;
background: #EF5A42;
cursor: pointer;
}
.nav-toggle span {
display: block;
```

```

position: absolute;
top: 19px;
left: 8px;
right: 8px;
height: 2px;
background: white;
}
.nav-toggle span:before, .nav-toggle span:after {
content: "";
position: absolute;
display: block;
left: 0;
width: 100%;
height: 2px;
background: white;
}
.nav-toggle span:before {
top: -10px;
}
.nav-toggle span:after {
bottom: -10px;
}
/* класс, который будет добавлен в верхнему меню при нажатии на кнопку и
покажет скрытое меню*/
#menu.active {
max-height: 123px;
}

```

## 8. Стили для блока с основным содержимым

```

/* левый контейнер */
.posts-list {
margin-bottom: 30px;
width: 64%;
float: left;
}
/* блок для статьи */
.post {
margin-bottom: 35px;
}
.post-content p {
line-height: 1.5;
padding-bottom: 1em;
}
.post-image {
margin-bottom: 30px;
}
.category {
margin-bottom: 15px;
}
.category a {

```

```
color: #F8B763;
text-transform: uppercase;
}
.post-title {
margin-bottom: 12px;
font-size: 26px;
}
/* блок с кнопкой "продолжить чтение" и кнопками социальных сетей */
.post-footer {
border-top: 1px solid #EBEBE3;
border-bottom: 1px solid #EBEBE3;
position: relative;
margin-top: 15px;
}
.more-link {
position: relative;
display: inline-block;
font-size: 10px;
text-transform: uppercase;
color: white;
line-height: 44px;
padding: 0 22px;
background: #3C3D41;
letter-spacing: 0.1em;
white-space: nowrap;
}
.more-link:after {
content: '';
display: block;
position: absolute;
width: 0;
height: 0;
top: 0;
right: 0;
border: solid transparent;
border-width: 22px 18px;
border-left-color: #3C3D41;
transform: translateX(100%);
}
.post-social {
position: absolute;
left: auto;
top: 50%;
right: 0;
text-align: right;
transform: translateY(-50%);
padding: 0;
font-size: 12px;
}
.post-social a {
display: inline-block;
margin-left: 8px;
```

```
color: #F8B763;
width: 25px;
height: 25px;
line-height: 23px;
text-align: center;
border-radius: 50%;
border: 1px solid;
}
```

## 9. Стили для боковой колонки

```
/* правый контейнер */
aside {
width: 33%;
float: right;
}
/* блок для виджетов */
.widget {
padding: 20px 15px;
background: white;
font-size: 13px;
margin-bottom: 30px;
box-shadow: 3px 3px 1px rgba(0,0,0,.05);
}
.widget-title {
font-size: 18px;
padding: 10px;
margin-bottom: 20px;
text-align: center;
border: 2px solid #F8B763;
box-shadow: 3px 3px 0 0 #F8B763;
}
.widget-category-list li {
border-bottom: 1px solid #EBEBE3;
padding: 10px 0;
color: #c6c6c6;
font-style: italic;
}
.widget-category-list li:last-child {
border-bottom: none;
}
.widget-category-list li a {
color: #626262;
margin-right: 6px;
font-style: normal;
}
.widget-category-list li a:before {
content: "\f105";
display: inline-block;
font-family: 'FontAwesome';
margin-right: 10px;
}
```

```
color: #c6c6c6;
}
.widget-posts-list li {
border-top: 1px solid #EBEBE3;
padding: 15px 0;
}
.widget-posts-list li:nth-child(1){
border-top: none;
}
.post-image-small {
width: 30%;
float: left;
margin-right: 15px;
}
.widget-post-title {
float: left;
}
/* форма подписки */
#subscribe {
position: relative;
width: 100%;
padding: 15px 0;
}
#subscribe input {
width: 100%;
display: block;
float: left;
border: 2px solid #EBEBE3;
padding: 0 0 0 10px;
height: 40px;
position: relative;
outline: none;
color: #9E9C9C;
font-style: italic;
}
#subscribe button {
padding: 0 15px;
background: transparent;
height: 40px;
border: none;
position: absolute;
right: 0;
color: #EF5A42;
cursor: pointer;
font-size: 18px;
}
#subscribe input:focus + button {
background: #EF5A42;
color: white;
}
```

## 10. Стили для нижнего колонтитула

Подвал сайта разделим на три равных столбца:

```
footer {
padding: 30px 0;
background: #3C3D41;
color: white;
}
.footer-col {
width: 33.3333333333%;
float: left;
}
.footer-col a {
color: white;
}
.footer-col:last-child {
text-align: right;
}
.social-bar-wrap {
text-align: center;
}
.social-bar-wrap a {
padding: 0 7px;
font-size: 18px;
}
```

## 11. Медиа-запросы

```
@media (max-width: 768px) {
/* показываем кнопку для переключения верхней навигации */
.nav-toggle {
display: block;
}
header {
padding: 10px 0;
}
/* скрываем верхнее меню, отменяем обтекание, позиционируем его, сместив
на высоту шапки сайта */
#menu {
max-height: 0;
background: white;
float: none;
position: absolute;
overflow: hidden;
top: 63px;
right: 0;
left: 0;
margin: 0;
padding: 0;
z-index: 3;
}
```



```
/* делаем элементы списка блочными, чтобы они располагались друг под
другом */
#menu li {
display: block;
padding-left: 15px;
border-bottom: 1px solid #E6E6E3;
margin-right: 0;
}
/* отменяем обтекание левой и правой колонок, устанавливаем им ширину
100%*/
.posts-list, aside {
width: 100%;
float: none;
}
.widget-post-title {
font-size: 1.5em;
}
}
@media (max-width: 480px) {
/* выравниванием логотип по центру */
nav {
text-align: center;
}
/* отменяем обтекание для логотипа */
.logo {
float: none;
margin-bottom: 15px;
}
/* позиционируем меню на увеличившуюся высоту шапки */
#menu {
top: 118px;
}
/* позиционируем форму поиска по левому краю */
#searchform {
float: left;
margin-left: 0;
}
/* убираем верхнюю и нижнюю границы и выравниваем кнопку по центру */
.post-footer {
border-top: none;
border-bottom: none;
text-align: center;
}
/* отменяем позиционирование кнопок соцсетей */
.post-social {
position: static;
text-align: center;
transform: none;
margin-top: 20px;
}
.widget-post-title {
font-size: 1.2em;
}
```

```

}
/* отменяем обтекание для столбцов подвала страницы */
.footer-col {
float: none;
margin-bottom: 20px;
width: 100%;
text-align: center;
}
.footer-col:last-child {
text-align: center;
margin-bottom: 0;
}
}

```

## 12. Скрипт для мобильного меню

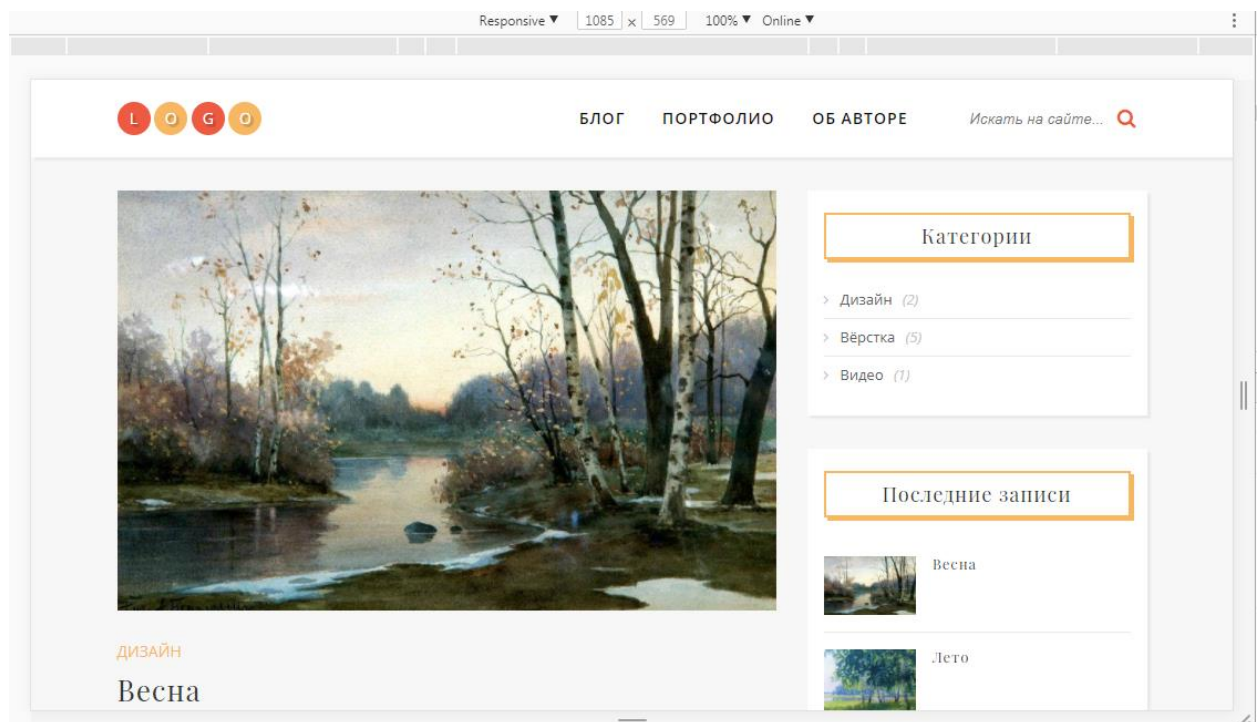
За показ-скрытие верхнего меню при клике на кнопку (переключается высота меню — от нулевой до равной её содержимому) отвечает код jQuery, который мы ранее добавили в разметку страницы перед закрывающим тегом `</body>`:

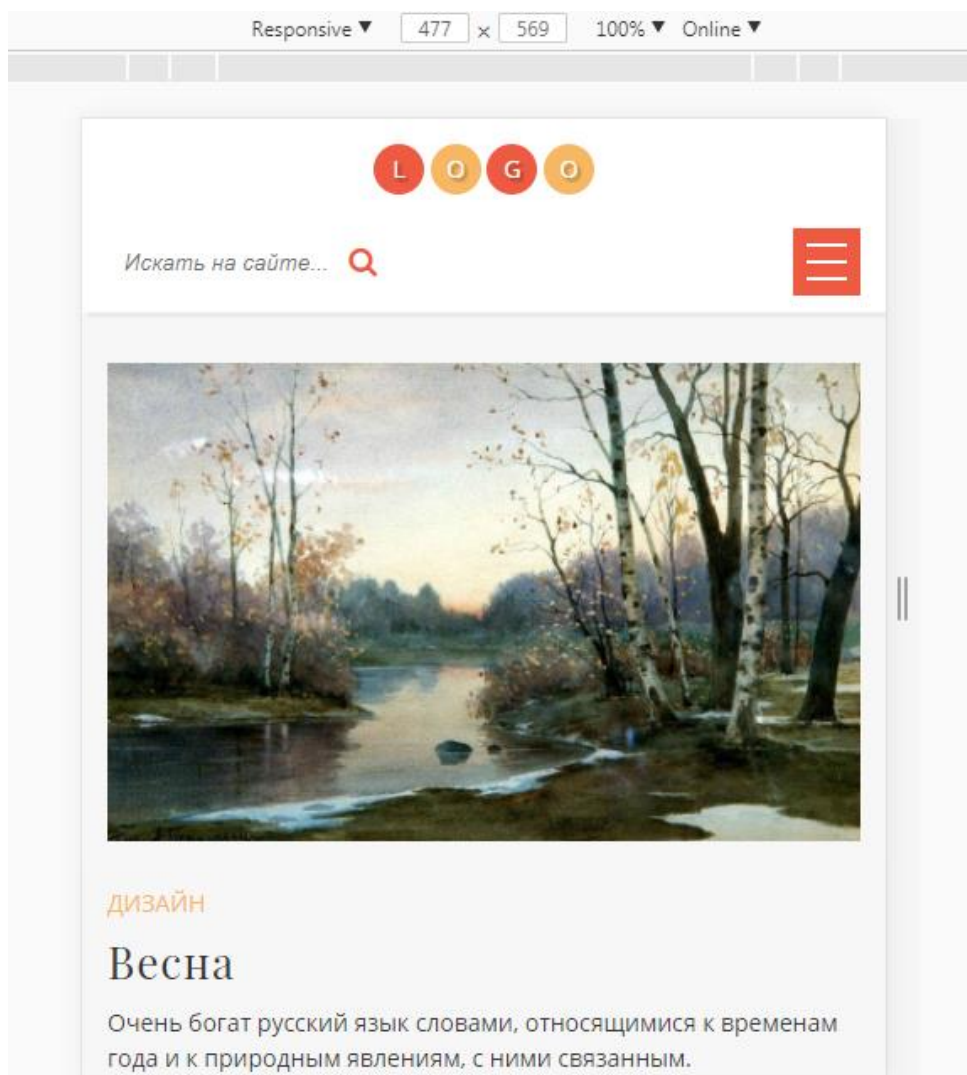
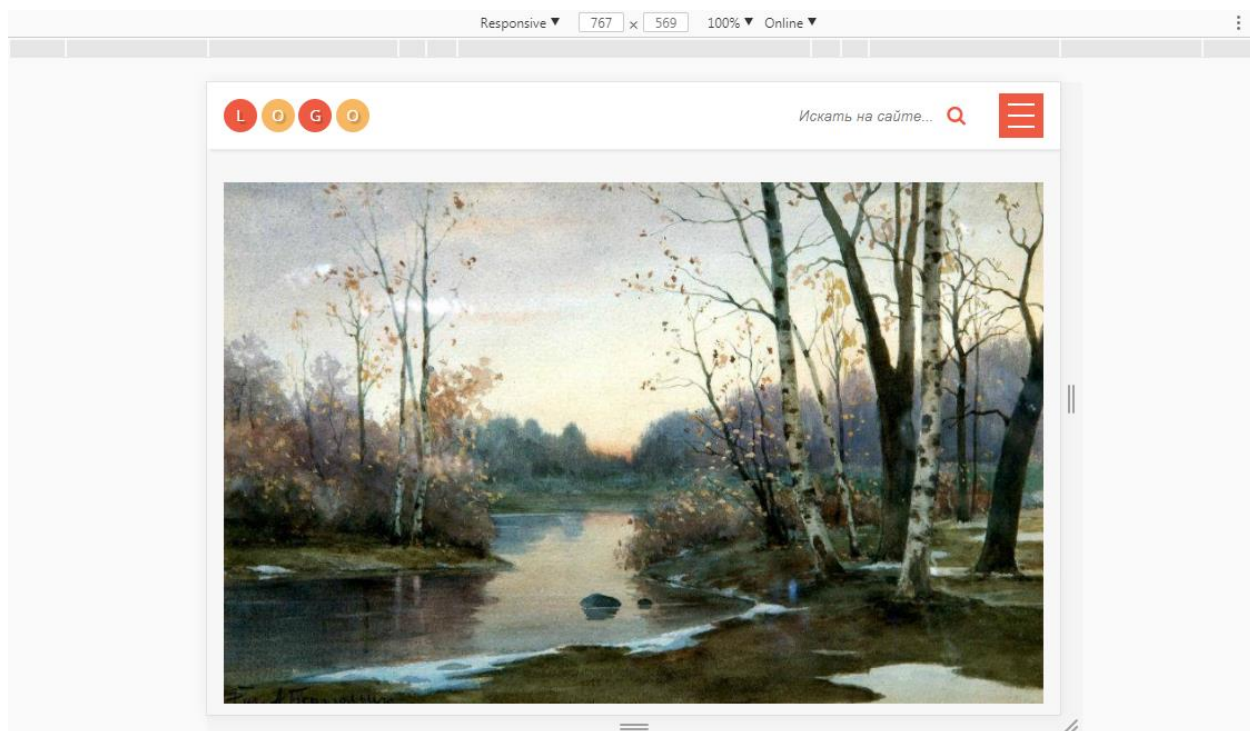
```

<script>
$('.nav-toggle').on('click', function(){
    $('#menu').toggleClass('active');
});
</script>

```

Результат:





#### **14. Валидация HTML и CSS.**

Для валидации используем сайты: <http://validator.w3.org>,  
<http://jigsaw.w3.org/css-validator>

#### **15. Выбор домена. Публикация сайта.**

Вам понадобится: Домен, хостинг, ftp-клиент

Хостинг – это сервис, предоставляющий вашему сайту определенный объем места на сервере. В зависимости от того, платный хостинг или бесплатный, размер этого места будет различаться. Платный хостинг обладает рядом неоспоримых преимуществ по сравнению с бесплатным – за исключением того, что за него нужно платить. Если вы выберете бесплатный хостинг, придется смириться с серьезными ограничениями в форматировании и дизайне страниц, а также с ограничением в использовании разных способов веб-программирования. Кроме того, на сайтах с бесплатным хостингом хостингпровайдеры по умолчанию размещают рекламу.

Если вы только начинаете свой путь в создании и размещении сайтов, бесплатный хостинг может стать хорошей отправной точкой. Его предоставляют различные сервисы – Narod.ru, Ucoz.ru, Boom.ru, и другие. В дальнейшем же стремитесь воспользоваться услугами платного хостинга.

После того, как вы выбрали хостинг, придумайте для вашего сайта доменное имя. Оно может состоять из букв, чисел и некоторых символов. На бесплатном хостинге вы получите доменное имя третьего уровня – оно должно быть уникальным и указывается чаще всего при регистрации в системе. Пример – name.narod.ru. Если же вы хотите зарегистрировать доменное имя второго уровня (name.ru), вам придется зарегистрироваться в системе платного хостинга и оплатить использование своего домена. Тарифы у каждого хостера различаются, и вы можете выбрать такой сервис, который наиболее устроит вас по условиям сотрудничества, технической поддержки и финансовой составляющей. Если вы планируете открывать сайт какого-то серьезного проекта или солидной фирмы, лучшим вариантом будет покупка доменного имени второго уровня – такой адрес сайта будет воспринят более серьезно, нежели адрес бесплатного хостинга.

После того, как домен выбран, и вы уже готовы к размещению и наполнению сайта, озаботьтесь загрузкой на сайт всего необходимого контента. Для этого можно использовать имеющийся интерфейс хостинг-системы, а можно установить на свой компьютер программу для загрузки файлов на сервер. Пример такой программы – CuteFTP. Скачайте программу, установите ее и зайдите в настройки файлового менеджера. Нажмите «Добавить сайт» и пропишите адрес сервера, на который вы будете загружать файлы. Также укажите логин и пароль, с помощью которых вы будете работать с сайтом через FTP-клиент. После того, как загрузка файлов на сервер будет завершена, ваш сайт появится на просторах всемирной паутины.

**Вы можете воспользоваться:**

– <https://ru.000webhost.com>

- <https://beget.ru/free-hosting>
- <https://byet.host/>
- <http://www.hostinger.in/>
- <http://www.hostinger.ru/>
- <http://Ohosting.me/>
- <http://nazuka.net/>
- <https://cloud.megaindex.ru/>
- <http://bestof.ru/>
- <http://www.hut.ru/>
- <http://oxnull.net/>
- <http://Owebhost.ru/>
- Примеры для методических указаний выполнены на хостинге <https://beget.ru/free-hosting> .

## **Регистрация на beget.ru**

Изначально производится регистрация в сервисе. Для этого осуществляется переход в раздел создания аккаунта <https://beget.ru/free-hosting> .

После этого, заполняется форма регистрации, как показано на рисунке 1.1, и нажимается кнопка “Зарегистрировать аккаунт”.

В открывшемся диалоговом окне нажимается кнопка “Готов получить код” (рисунок 1.2), далее следует подтвердить регистрацию кодом, введя полученные цифры из СМС и нажав кнопку “Завершить регистрацию”, как показано на рисунке 1.3.

## Получить бесплатный хостинг

ФИО \*

Страна и моб. телефон \*

Россия ▼ +7

☒ с [соглашением использования сервиса](#) ознакомлен, все условия принимаю

Зарегистрировать аккаунт

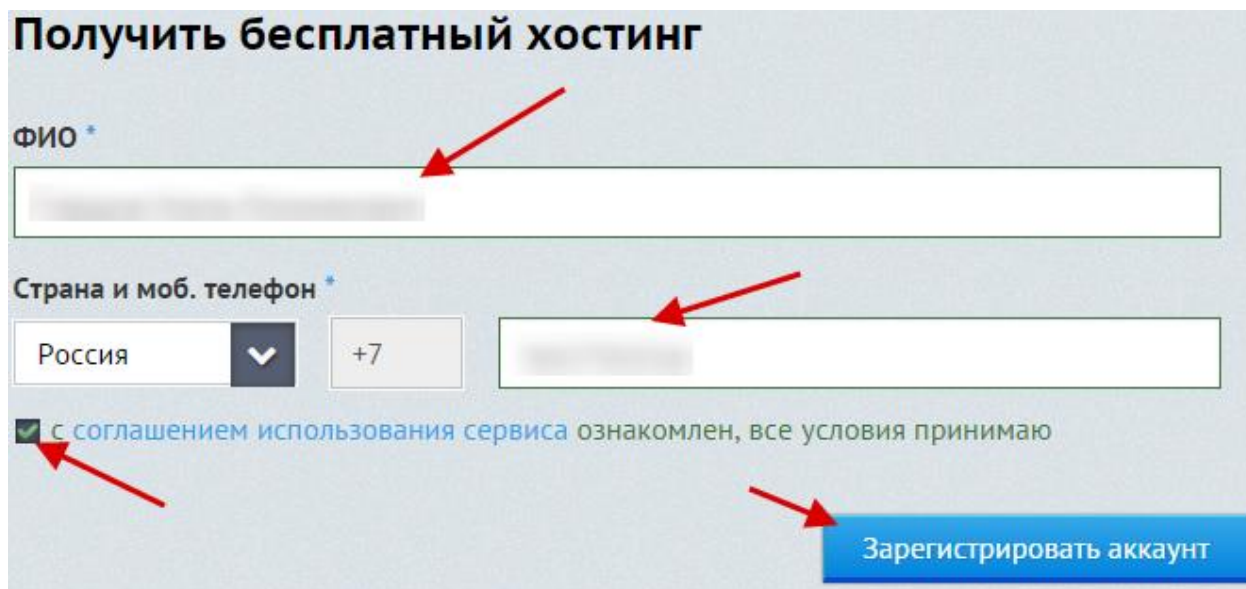


Рисунок 1.1 - Форма регистрации

### Подтверждение телефонного номера

Для завершения регистрации Вам необходимо ввести код подтверждения, который Вы получите в SMS-сообщении.

Ваш номер телефона: +7

Готов получить код

Вернуться к форме

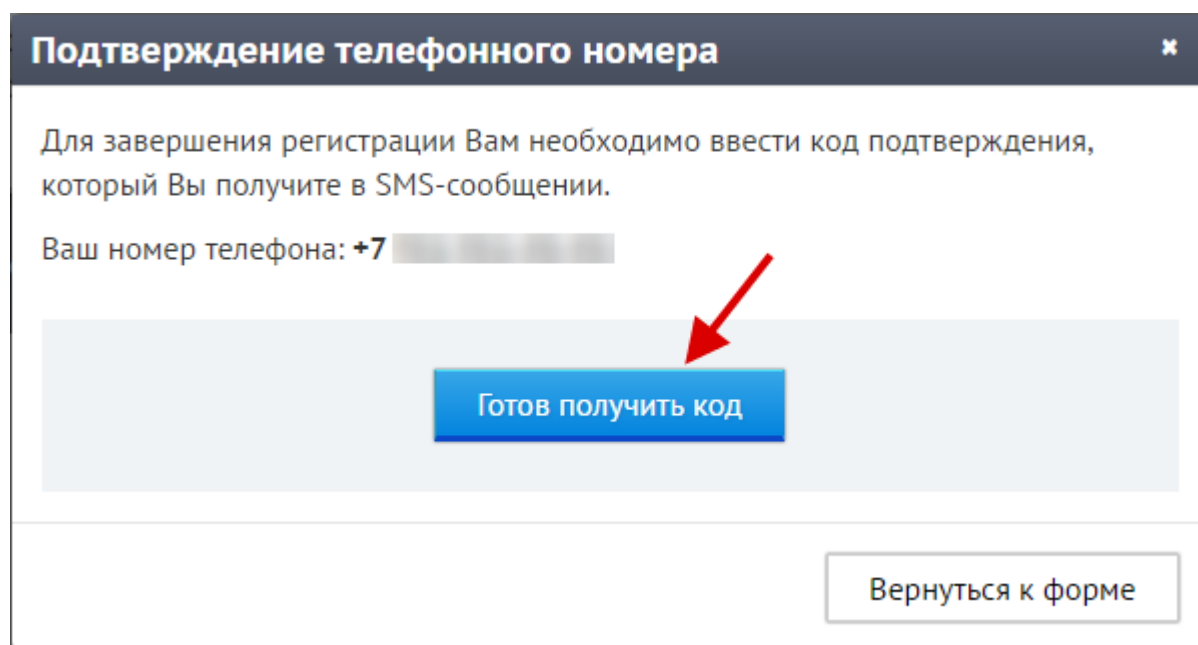


Рисунок 1.2 - Подтверждение готовности получения СМС с кодом



**Подтверждение телефонного номера**

Для завершения регистрации Вам необходимо ввести код подтверждения, который Вы получите в SMS-сообщении.

Ваш номер телефона: +7 [blurred number]

Код подтверждения: [input field with blurred code] [Повторная отправка](#)

[Вернуться к форме](#) [Завершить регистрацию](#)

Рисунок 1.3 - Подтверждение регистрации

После подтверждения регистрации, откроется страница с данными для входа в панель управления (<https://cp.beget.ru/login.php>). Производится нажатие на кнопку “Начать работу”, как показано на рисунке 1.4, и осуществится переход в панель управления, как показано на рисунке 1.5.

Возможно разместить файлы на хостинге посредством файлового менеджера (ярлык файлового менеджера в панели управления показан на рисунке 1.6).

Данные для доступа к Вашему серверу и "Панели Управления Аккаунтом" (ПУА):

- > Адрес ПУА: <https://cp.beget.ru>
- > Имя пользователя: [blurred]
- > Пароль: [blurred]
- > Сервер: [blurred]

[Начать работу](#)

Ознакомьтесь с инструкцией по нашей [Панели Управления](#). (откроется в новом окн)

Рисунок 1.4 - Завершение регистрации

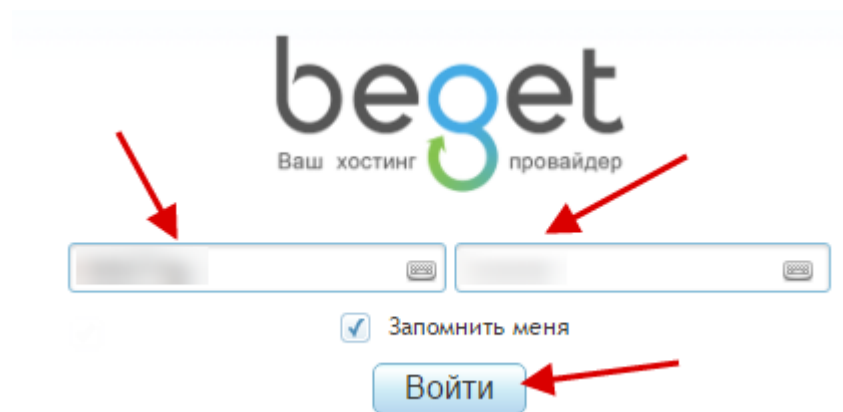


Рисунок 1.5 - Вход в панель управления

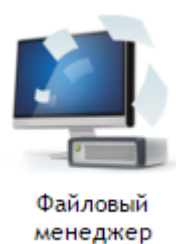


Рисунок 1.6 - Ярлык файлового менеджера в панели управления хостинга

В файловом менеджере для загрузки файлов используется кнопка “Загрузить Файлы”, как показано на рисунке 1.7. Так же в файловом менеджере возможно увидеть адрес сайта, который привязан к аккаунту.

В качестве корневой папки, доступной пользователям сайта и куда следует загружать все файлы, используется “public\_html”.

После нажатия на кнопку “Загрузить Файлы”, откроется окно загрузки файлов, как показано на рисунке 1.8.



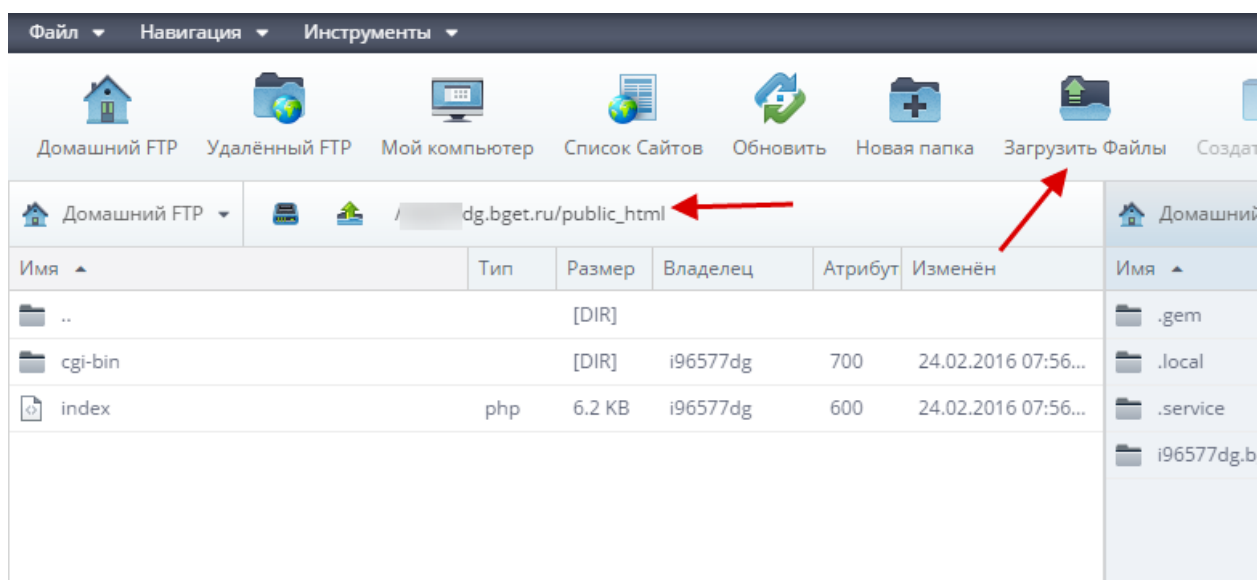


Рисунок 1.7 - Файловый менеджер хостинга

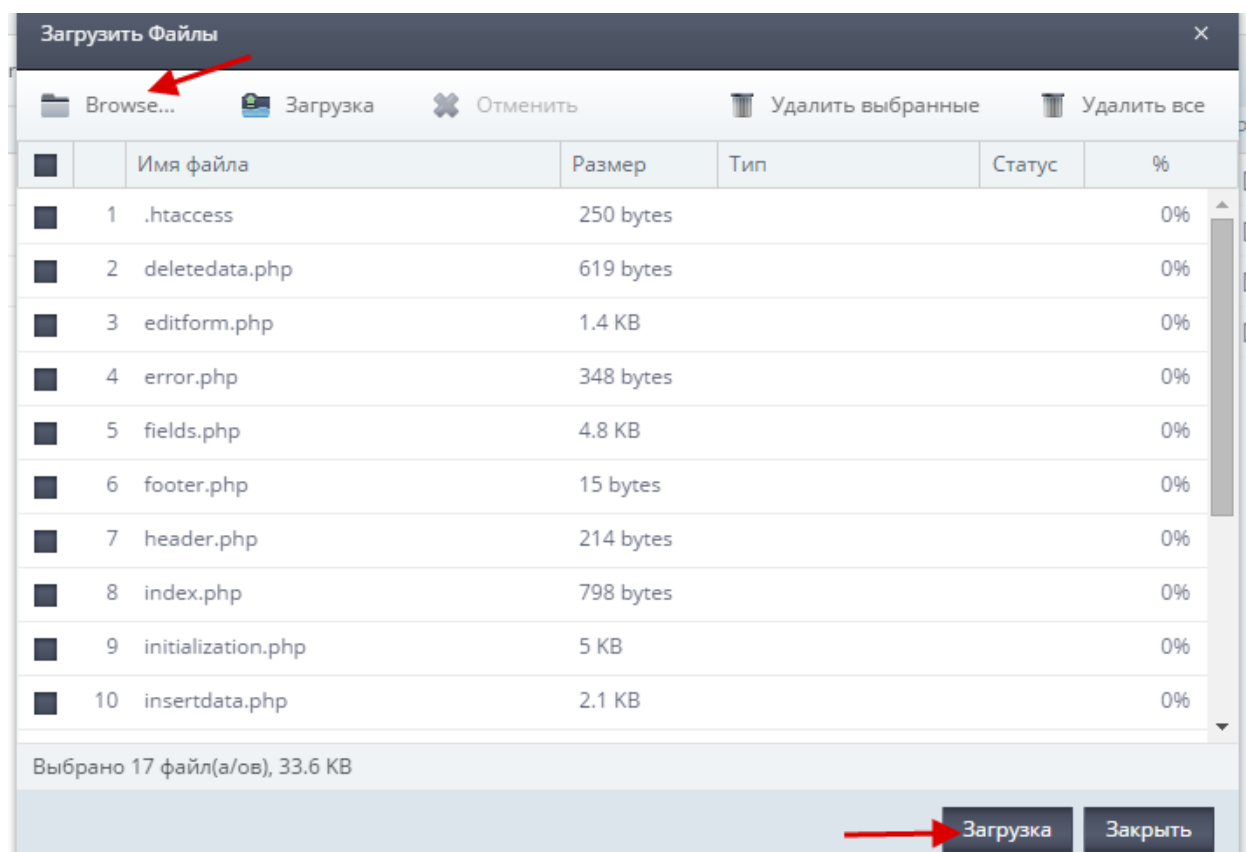


Рисунок 1.8 - Окно загрузки файлов файлового менеджера хостинга

Адрес сайта возможно узнать перейдя в раздел “Домены” панели администрирования (рисунок 1.9). Адрес сайта отображен в таблице доступных доменов (рисунок 1.10).



Домены

Рисунок 1.9 - Раздел “Домены”



#### Бонусные домены

У Вас нет бонусных доменов.

[Как их получить?](#)

#### Зарегистрировать новый домен

[У меня уже есть домен](#)

##### Ручное продление домена

Возможность продлить домен вручную появляется за 60 дней до конца периода делегирования домена. Домен будет продлен за счет средств на счету аккаунта (это может уменьшить количество дней хостинга со скидкой, если у вас оплачен годовой тариф) или за счет бонусного домена, если он есть.

##### Автоматическое продление домена

Домены, для которых было включено автоматическое продление, будут продлеваться на следующий год автоматически при условии наличия достаточных средств на счету (это может уменьшить количество дней хостинга со скидкой, если у вас оплачен годовой тариф) или бонусных доменов.

Обратите внимание, что автопродление начинает работать за 2 месяца до конца срока делегирования домена.

##### Перенос домена на обслуживание

Вы можете перенести домен к нам на обслуживание, чтобы иметь возможность продлевать его из нашей панели управления аккаунтом.

Домен	Период делегирования	Фильтр
i dg.bget.ru	бессрочно	

Рисунок 1.10 - Адрес сайта

## Практическое задание

1. Выбрать тему. Актуализировать ее.
2. Проанализировать сайты по аналогичной тематике
3. Собрать, систематизировать материал и создать структуру сайта
4. Создать дизайн макета главной страницы
5. Создать шапку сайта
6. Создать выпадающее меню
7. Создать подвал сайта
8. Наполнить сайт контентом. Создать ссылки
9. Добавить рисунков, аудио и видео-контента на сайт

10. Создать слайдер
11. Создать форм регистрации и поиска по сайту
12. Создать таблицу
13. Сделать адаптивную верстку с использованием CSS
14. Провести валидацию HTML и CSS файлов
15. Проверить на кроссбраузерность сайт
16. Выбрать домен и хостинг. Опубликовать сайт.
17. Создать канал и опубликовать мультимедиа контент
18. Создать плейлист. Реализовать перекрестные ссылки между каналом и сайтом
19. Оформить отчет