

Веб-разработка

Что такое веб-разработка?

Веб-разработка — процесс создания веб-сайта или веб-приложения.

Для начала давайте разберёмся, что такое веб-сайт и рассмотрим процесс создания веб-сайта.

Сайт или **веб-сайт** (от англ. website: web — «паутина, сеть» и site — «место», буквально «место, сегмент, часть в сети») - одна или несколько логически связанных между собой веб-страниц. Обычно сайт в Интернете представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователями как единое целое. Веб-сайты называются так, потому что доступ к ним происходит по протоколу HTTP.

Все сайты в совокупности составляют Всемирную паутину, которая объединяет информацию мирового сообщества в единое целое — базу данных и паутину планетарного масштаба.



Когда появился первый сайт?

Первый в мире сайт - **info.cern.ch**, он появился 6 августа 1991 года. Его создателем является Тим Бернерс-Ли. Он опубликовал на нём описание новой технологии World Wide Web, основанной на протоколе передачи данных HTTP,

системе адресации URI и языке гипертекстовой разметки HTML. Также на сайте были описаны принципы установки и работы серверов и браузеров.

Кроме того, Тим Бернерс-Ли разместил на нём список ссылок на другие сайты, и сайт стал ещё и первым в мире интернет-каталогом.

Тим Бернерс-Ли считал, что гипертекст может служить основой для сетей обмена данными. Ещё в 1980 году он создал гипертекстовое программное обеспечение Enquire. Затем, работая в Европейском центре ядерных исследований в Женеве (CERN), он предложил коллегам публиковать гипертекстовые документы, связанные между собой гиперссылками. Бернерс-Ли продемонстрировал возможность гипертекстового доступа к внутренним поисковику и документам, а также новостным ресурсам Интернета. В результате, в мае 1991 года в CERN был утверждён стандарт WWW.

Тим Бернерс-Ли является «отцом» основополагающих технологий веба — HTTP, URI/URL и HTML, хотя их теоретические основы были заложены ещё раньше.

HyperText Transfer Protocol

Для прямого доступа клиентов к сайтам на серверах был специально разработан протокол HTTP.

HTTP — широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (то есть документов, которые могут содержать ссылки, позволяющие организовать переход к другим документам).

Аббревиатура HTTP расшифровывается как HyperText Transfer Protocol, «протокол передачи гипертекста». В соответствии со спецификацией OSI, HTTP является протоколом прикладного (верхнего, 7-го) уровня.

Модель открытых систем OSI					
Уровень (layer)		Тип данных	Функции уровня	Особенность адресации	Примеры протоколов
Уровни хоста (узла)	7. Прикладной (application)	Данные (строки из байтов)	Доступ к сетевым службам	URL	HTTP(S), FTP(S), RPC, POP3
	6. Представительский (представления) (presentation)		Представление (кодировка) и шифрование данных		ASCII, EBCDIC
	5. Сеансовый (session)		Управление сеансом связи		PAP
	4. Транспортный (transport)	Сегменты (segment) / Дейтаграммы (datagram)	Прямая связь между конечными пунктами и надёжность	Порт	TCP, UDP, SCTP, PORTS
Уровни связи (сети)	3. Сетевой (network)	Пакеты (packet)	Определение маршрута и логическая адресация	IP-адрес	IPv4, IPv6, IPsec, AppleTalk
	2. Канальный (data link)	Биты (bit) / Кадры (frame)	Физическая адресация	MAC-адрес (физический адрес компьютера)	PPP, IEEE 802.22, Ethernet, DSL, ARP, L2TP, сетевая карта.
	1. Физический (physical)	Электрические сигналы, Биты (bit)	Работа со средой передачи, сигналами и двоичными данными		USB, кабель ("витая пара", коаксиальный, оптоволоконный), радиоканал

Протокол HTTP предполагает использование клиент-серверной структуры передачи данных:

- 1) Клиентское приложение формирует запрос и отправляет его на сервер.
- 2) Серверное программное обеспечение обрабатывает данный запрос.
- 3) Серверное программное обеспечение формирует ответ и передаёт его обратно клиенту.

После этого клиентское приложение может продолжить отправлять другие запросы, которые будут обработаны аналогичным образом.

Главная задача, которую решает протокол HTTP — обмен данными между пользовательским приложением, осуществляющим доступ к веб-ресурсам. На данный момент благодаря протоколу HTTP обеспечивается работа Всемирной паутины. Также HTTP часто используется как протокол передачи информации для других протоколов прикладного уровня, таких как SOAP, XML-RPC и WebDAV. В таком случае говорят, что протокол HTTP используется как «транспорт». API многих программных продуктов также подразумевает использование HTTP для передачи данных — сами данные при этом могут иметь любой формат.

Как правило, передача данных по протоколу HTTP осуществляется через TCP/IP-соединения (Transmission Control Protocol (TCP) и Internet Protocol (IP)). Серверное программное обеспечение при этом обычно использует TCP-порт 80 (и, если порт не указан явно, то обычно клиентское программное обеспечение по умолчанию использует именно 80-й порт для открываемых HTTP-соединений), хотя может использовать и любой другой.

Интересный факт!

В 2006 году в Северной Америке доля HTTP-трафика превысила долю P2P-сетей и составила 46%, из которых почти половина — это передача потокового видео и звука.

Теперь давайте узнаем о языке HTML и стандарте CSS.

HTML (HyperTextMarkupLanguage -«язык гипертекстовой разметки») - стандартизированный язык разметки веб-страниц во Всемирной паутине. Код HTML интерпретируется браузерами; полученная в результате интерпретации страница отображается на экране монитора компьютера или мобильного устройства.

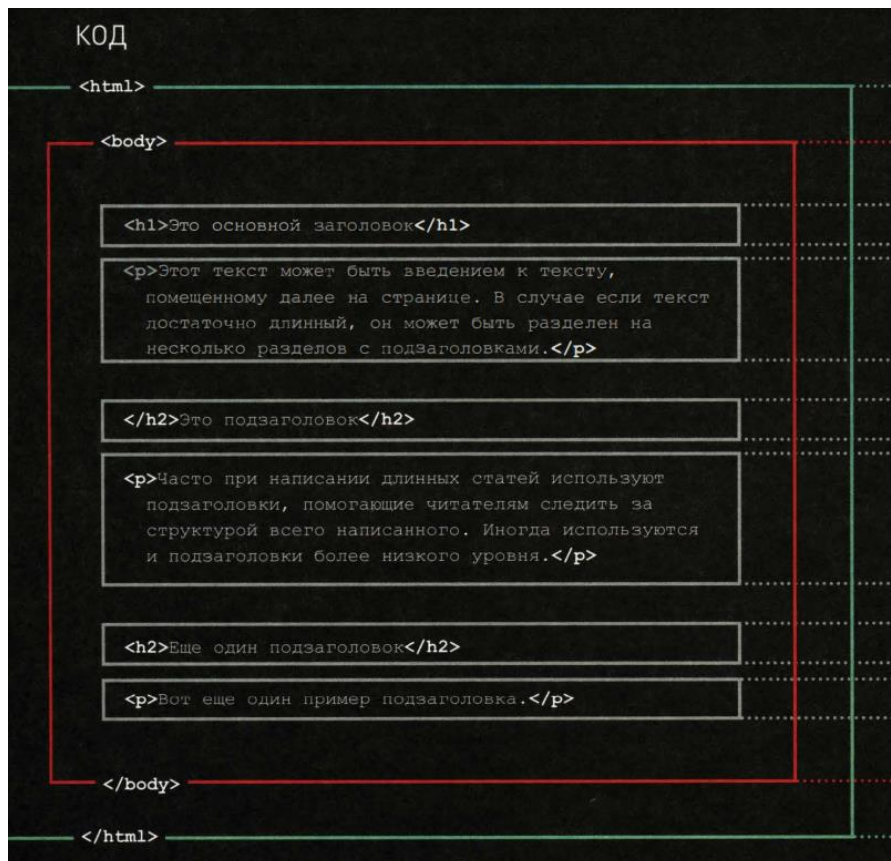
Язык гипертекстовой разметки был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986—1991 годах. С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста и мультимедийные возможности.

Существуют следующие версии HTML:

- HTML 2.0 – (24 ноября 1995 года);
- HTML 3.0 – (28 сентября 1995 года);
- HTML 3.2 – (14 января 1997 года);
- HTML 4.0 – (18 декабря 1997 года);
- HTML 4.01 – (24 декабря 1999 года);
- ISO/IEC 15445:2000 (так называемый ISO HTML, основан на HTML 4.01 Strict) – 15 мая 2000 года;

- HTML5 – (28 октября 2014 года);
- HTML 5.1 начал разрабатываться 17 декабря 2012 года. Рекомендован к применению с 1 ноября 2016 года;
- HTML 5.2 (14 декабря 2017 года);
- HTML 5.3 (24 декабря 2018 года);

Ниже приведен код на языке HTML:



```
КОД
<html>
  <body>
    <h1>Это основной заголовок</h1>
    <p>Этот текст может быть введением к тексту, помещенному далее на странице. В случае если текст достаточно длинный, он может быть разделен на несколько разделов с подзаголовками.</p>
    <h2>Это подзаголовок</h2>
    <p>Часто при написании длинных статей используют подзаголовки, помогающие читателям следить за структурой всего написанного. Иногда используются и подзаголовки более низкого уровня.</p>
    <h2>Еще один подзаголовок</h2>
    <p>Вот еще один пример подзаголовка.</p>
  </body>
</html>
```

Далее рассмотрим язык описания CSS.

CSS (от англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML).

История создания и развития CSS:

В 1990-х годах стала ясна необходимость стандартизировать Web, создать какие-то единые правила, по которым программисты и веб-дизайнеры проектировали бы сайты. Так появились языки HTML 4.01 и XHTML, и стандарт CSS.

В начале 1990-х различные браузеры имели свои стили для отображения веб-страниц. Так как HTML развивался очень быстро и был способен удовлетворить все существовавшие на тот момент потребности по оформлению информации, CSS не получил тогда широкого признания.

Предложил «каскадные таблицы стилей» Хоконом Ли в 1994 году. Он стал развивать CSS вместе с Бертом Босом. В отличие от многих существовавших на тот момент языков стиля, CSS использует наследование от родителя к потомку, поэтому разработчик может определить разные стили, основываясь на уже определённых ранее стилях.

В середине 1990-х Консорциум Всемирной паутины (W3C) стал проявлять интерес к CSS, и в декабре 1996 года была издана рекомендация CSS1.

Теперь CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц.

Основанная цель:

Основной целью разработки CSS являлось отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Цель языка CSS – увеличение доступности документа, его гибкости, а также уменьшение сложности и повторяемости в структурном содержимом.

Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана).

Существует 4 уровня CSS:

Уровень 1 (CSS1)

Была принята 17 декабря 1996 года (откорректирована 11 января 1999 года).

Возможности, предоставляемые этим уровнем:

- Параметры шрифтов. Возможности по заданию гарнитуры и размера шрифта, а также его стиля (обычный, курсивный или полужирный).
- Цвета. Спецификация позволяет определять цвета текста, фона, рамок и других элементов страницы.
- Атрибуты текста. Возможность задавать межсимвольный интервал, расстояние между словами и высоту строки.
- Выравнивание для текста, изображений, таблиц и других элементов.
- Свойства блоков, такие как высота, ширина, внутренние (padding) и внешние (margin) отступы и рамки. Также в спецификацию входили ограниченные средства по позиционированию элементов, такие как `float` и `clear`.

Уровень 2 (CSS2)

Была принята 12 мая 1998 года. В основе CSS1, добавление к функциональности:

- Блочная вёрстка. Появились относительное, абсолютное и фиксированное позиционирование (позволяет управлять размещением элементов по странице без табличной вёрстки).
- Типы носителей. Позволяет устанавливать разные стили для разных носителей (например монитор, принтер, КПК).
- Звуковые таблицы стилей. Определяет голос, громкость и т. д. для звуковых носителей (например для слепых посетителей сайта).
- Страничные носители. Позволяет, например, установить разные стили для элементов на чётных и нечётных страницах при печати.
- Расширенный механизм селекторов.
- Указатели.
- Генерируемое содержимое. Позволяет добавлять содержимое, которого нет в исходном документе, до или после нужного элемента.

Уровень 2, ревизия 1 (CSS2.1)

Была принята 7 июня 2011 года.

CSS2.1 основана на CSS2. В ней исправлены ошибки и изменены некоторые части спецификации, а некоторые и вовсе удалены.

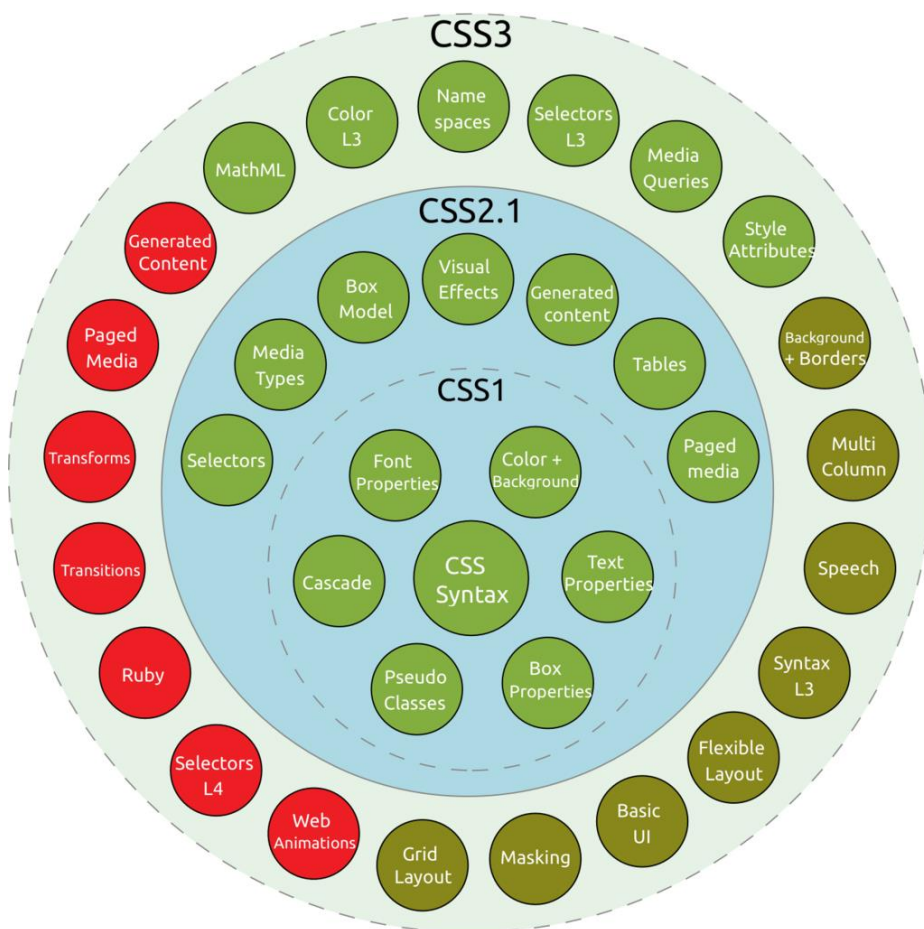
Уровень 3(CSS3)

CSS3 -Представляет собой формальный язык, реализованный с помощью языка разметки. . CSS3 основан на CSS2.1, но является самой масштабной редакцией по сравнению с CSS1, CSS2 и CSS2.1. Главной особенностью CSS3 является возможность создавать анимированные элементы без использования JavaScript, поддержка линейных и радиальных градиентов, теней, сглаживания и прочее.

Кроме того, спецификация разбита на модули, разработка и развитие которых идёт независимо.

Уровень 4(CSS4)

Разрабатывается W3C с 29 сентября 2011 года. Модули CSS4 построены на основе CSS3 и дополняют их новыми свойствами и значениями. Они существуют пока в виде черновиков.



Этапы разработки веб-сайтов:

На сегодняшний день существуют несколько этапов разработки веб-сайта:

- 1) Проектирование сайта или веб-приложения (сбор и анализ требований, разработка технического задания, проектирование интерфейсов);
- 2) Разработка креативной концепции сайта;
- 3) Создание дизайн-концепции сайта:

Начинается работа с создания дизайна, обычно в графическом редакторе. Дизайнер создаёт один или несколько вариантов дизайна, в соответствии с техническим заданием. При этом отдельно создаётся дизайн главной страницы, и дизайны типовых страниц (например: статьи, новости, каталог продукции). Собственно «дизайн страницы» представляет собой графический файл, слоеный рисунок, состоящий из наиболее мелких картинок-слоев элементов общего рисунка.

При этом дизайнер должен учитывать ограничения стандартов HTML (не создавать дизайн, который затем не сможет быть реализован стандартными средствами HTML). Исключение составляет Flash-дизайн. Количество эскизов и порядок их предоставления оговаривается с проект-менеджером. Также менеджер проекта осуществляет контроль сроков. В больших веб-студиях в процессе участвует арт-директор, который контролирует качество графики. Этап также заканчивается утверждением эскиза заказчиком.

- 4) Создание макетов страниц;
- 5) Создание мультимедиа и FLASH-элементов;
- 6) Вёрстка страниц и шаблонов
- 7) HTML-верстка

Утверждённый дизайн передаётся HTML-верстальщику, который «нарезает» графическую картинку на отдельные рисунки, из которых впоследствии складывается HTML-страницу. В результате создаётся код, который можно просматривать с помощью браузера. А типовые страницы впоследствии будут использоваться как шаблоны.

Именно благодаря вёрстке макеты сайта становятся веб-страницами.

8) Программирование (разработка функциональных инструментов) или интеграция в систему управления содержимым (CMS);

Далее готовые HTML-файлы передают программисту. Программирование сайта может осуществляться как «с нуля», так и на основе CMS — системы управления сайтом. Веб-разработчики часто называют CMS «движком».

В случае с CMS надо сказать, что сама «CMS» в некотором смысле это готовый сайт, состоящий из заменяемых частей. «Программист» — в данном случае правильно будет назвать его просто специалистом по CMS — должен заменить стандартный шаблон, поставлявшийся с CMS, на оригинальный шаблон. Этот оригинальный шаблон он и должен создать на основе исходного «веб-дизайна». При программировании сайта специалисту назначаются контрольные точки сроков.

9) Оптимизация и размещение материалов сайта;

10) Тестирование и внесение корректировок;

Процесс тестирования может включать в себя самые разнообразные проверки: вид страницы с увеличенными шрифтами, при разных размерах окна браузера, при отсутствии флэш-плеера и многие другие. Также — юзабилити-тестирование. Обнаруженные ошибки отправляются на исправление до тех пор, пока не будут устранены. Сроки контролирует менеджер проекта. Также, на этом этапе привлекают к работе дизайнера, чтобы он провёл авторский надзор.

11) Публикация проекта на хостинге;

Файлы сайта размещают на сервере провайдера (хостинга) и производят нужные настройки. На этом этапе сайт пока закрыт для посетителей.

12) Обслуживание работающего сайта или его программной основы.

Создадим свою простейшую веб-страницу?

- 1) Запустим программу Brackets;
- 2) Теперь пишем «скелет» сайта index1.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Моя первая веб-страница</title>
<link rel="stylesheet" href="main.css">
</head>
<body>
  <div class="header">
    <a href="#"> Главная </a>
    <a href="pz.html"> Игра в загадки </a>
    <a href="guess.html"> Угадайка </a>
  </div>
  <h1>Личный сайт студента МАИ </h1>
<div class="content">
  
<div class="center">
  <p><b> Добрый день!</b></p>
  <p>Меня зовут <i>Кузнецова Дарина.</i></p>
  <p><b>И это мой сайт для реферата.</b></p>
  <p>Вот моя страница <a href="https://vk.com/kuzzzznetsova">Вконтакте.</a> </p>
  <p>На этом сайте вы сможете сыграть в несколько игр, которые я написала.</p>
</div>
</div>
<br> <br>
<div class="footer">
  <a href="#"> Главная </a>
  <a href="pz.html"> Игра в загадки </a>
  <a href="guess.html"> Угадайка </a>
<br><br>
  Copyright &copy; Darina Kuznetsova
</div>
</body>
</html>
```

- 3) Добавим оформление с помощью main.css :

```
body {
  background-color: beige;
  font-size: 14px;
}
h1 {
  color:red;
}
.header{
  border-width: 0.2px;
  border-color: #4d4d4d;
  border-style: solid;
  background-color: #99CCCC;
  height: 48px;
  text-align : center;
  padding-top: 20px;
```

```

}
.header a{
  color: #4d4d4d;
  font-size: 20px;
  font-weight: 100;
  padding-right: 25px;
}

.footer {
  border-width: 0.2px;
  border-color: #4d4d4d;
  background-color: #669999;
  height: 80px;
  padding-top: 60px;
  text-align: center;
  padding-top: 60px;
  position: absolute;
width: 100%;
height: 60px;
bottom: 0px;
}
.footer a {
  color: #CCFFFF;
  padding-right: 25px;
  font-size: 20px;
  font-weight: 100;
}

h1 {
  text-align: center;
  color: #003333;
}

img {
  width: 300px;
}

.center {
  border-width: 0.1px;
  border-color: ghostwhite;
  border-style: solid;
  background-color: white;
  height: 230px;
  width: 1000px;
  text-align: center;
  padding-top: 10px;
  float: right;
  margin-bottom: 100px;
}

.center p{
  font-size: 18px;
}

.content{
padding-left: 10px;
padding-right: 100px;
}

```

```
.box a {
  color: #4d4d4d;
  font-size: 30px;
}
.box p {
  color: black;
  font-size: 18px;
}
```

4) Напишем дополнительные коды для игр «Угадайка» и «Игра в загадки».

pz.html («Игра в загадки»)

guess.html («Угадайка»)

Для игры «Угадайка» воспользуемся тегами `<script>` и `</script>`, чтобы вставить в файл html файл js.

«Угадайка»:

```
<!DOCTYPE html>
<html>

<head>
<meta charset="utf-8">
<title>Моя первая веб-страница</title>
<link rel="stylesheet" href="main.css">
<script type="text/javascript">

function guess() {
  var answer = parseInt(Math.random() * 100);
  var userAnswer = + prompt ("Угадайте число от 0 до 100");
  var maxTryCount = 5;

  for (var tryCount = 1; tryCount < maxTryCount; tryCount++) {
    if(userAnswer == answer) {
      alert("Поздравляю, вы угадали!");
      break;
    } else if (userAnswer > answer) {
      alert("Вы ввели слишком большое число.");
    } else if (userAnswer < answer) {
      alert("Вы ввели слишком маленькое число.");
    }
    userAnswer = +prompt ("Попробуйте ещё раз. Введите число от 1 до 100\n У вас
осталось " + (maxTryCount - tryCount) + " попыток");
  }
  if (tryCount == maxTryCount) {
    alert("Вы проиграли!");
  }
  alert("Правильный ответ: " + answer);
}
</script>
</head>

<body>

<div class="header">
  <a href="index1.html"> Главная </a>
```

```
    <a href="pz.html"> Игра в загадки </a>
    <a href="guess.html"> Угадайка </a>
</div>
```

```
    <h1>Игра Угадайка</h1>
    <div class="box">
        <a href="#" onclick="guess();">Начать</a>
    </div>
    <br>
    <br>
```

```
    <div class="footer">
        <a href="index1.html"> Главная </a>
        <a href="pz.html"> Игра в загадки </a>
        <a href="guess.html"> Угадайка </a>

        <br><br>
        Copyright &copy; Darina Kuznetsova
    </div>
```

```
</body>
```

```
</html>
```

«Игра в загадки»:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Моя первая веб-страница</title>
```

```
<link rel="stylesheet" href="main.css">
```

```
<script type="text/javascript">
```

```
    function checkAnswers(){
```

```
        var userAnswer = document.getElementById("userAnswer1").value;
```

```
        var score = 0;
```

```
        userAnswer = userAnswer.toLocaleLowerCase();
```

```
        if(userAnswer == "сон" || userAnswer == "сноведение"){
```

```
            score++;
```

```
        }
```

```
        var userAnswer = document.getElementById("userAnswer2").value;
```

```
        userAnswer = userAnswer.toLocaleLowerCase();
```

```
        if(userAnswer == "лестнице" || userAnswer == "по лестнице" || userAnswer==
"лестница"){
```

```
            score++;
```

```
        }
```

```
        var userAnswer = document.getElementById("userAnswer3").value;
```

```
        userAnswer = userAnswer.toLocaleLowerCase();
```

```
        if(userAnswer == "в словаре" || userAnswer == "словарь") {
```

```
            score++;
```

```
        }
```

```
        alert("Всего отгадано: " + score);
```

```
    }
```



```

</script>
</head>

<body>

<div class="header">
  <a href="index1.html"> Главная </a>
  <a href="#"> Игра в загадки </a>
  <a href="guess.html"> Угадайка </a>
</div>

<h1>Игра в загадки</h1>

<div class="box">
  <p> Что можно увидеть с закрытыми глазами? </p>
  <input type="text" id="userAnswer1">
  <p> По чему ходят часто, а ездят редко? </p>
  <input type="text" id="userAnswer2">
  <p> Где вторник идёт раньше, чем понедельник? </p>
  <input type="text" id="userAnswer3">
  <br> <br> <br>
  <a href="#" onClick="checkAnswers();" ><b>Ответить</b></a>
</div>
<br>
<div class="footer">
  <a href="index1.html"> Главная </a>
  <a href="#"> Игра в загадки </a>
  <a href="guess.html"> Угадайка </a>
  <br><br>
  CopyRight &copy; Darina Kuznetsova
</div>
</body>
</html>

```

5) Не забываем добавлять ссылки index1.html , pz.html, guess.html.

6) Сохраняем эти файлы, выбираем пункт «Все файлы» (All Files) в раскрывающемся списке «Тип файла» (Save as type), чтобы сохранить документ как веб-страницу, а не как простой текст.

Для веб-страниц с кириллическими буквами обязательно выбираем пункт UTF-8 в раскрывающемся списке «Кодировка» (Encoding). Это необходимо для правильного отображения русского текста в окне браузера.

Заключение:

В наше время почти каждый человек проводит много времени в Интернете. Главной задачей веб-разработчиков является создание веб-сайтов, в которых присутствует не только интересный контент, но и в которых сочетается гармоничный дизайн и удобный интерфейс. Для этого существуют современные языки создания сайтов и редакторы для них. При написании данного реферата были проанализированы язык создания сайтов html и редактор создания сайтов css. Кроме того, разработан личный сайт. Разработанный сайт предоставляет пользователю возможность поиграть в игры, написанные с помощью языка программирования Java Script.