

# Курс «Основы веб-программирования»

## Лабораторная 1. HTML

Целью данной лабораторной является создание двух веб-страниц: с информацией об авторе и о развитии проекта.

### Порядок работы

1. Создай страницу с названием **"Об авторе"**.
2. Размести на странице пару любимых цитат.
3. Создай на странице раздел **"Коротко обо мне"** и помести в него небольшой параграф с информацией о себе.  
**ВВ!** Указывать реальные сведения необязательно.
4. Создай раздел **"Увлечения"** и размести в нём нумерованный список с наиболее любимыми занятиями.
5. Создай раздел **"Текущие проекты"** и помести в нём ссылку на страницу курса "Основы веб-программирования".
6. Помести картинку в самом начале страницы. Укажи для неё альтернативный текст с описанием.
7. Создай отдельную страницу о развитии своего проекта. Дай ей подходящий заголовок и название.
8. Создай раздел **"Дневник разработки"** с подразделами **"План действий"** и **"Лабораторная 1. HTML и поиск"**.
9. Помести в раздел **"План действий"** таблицу следующего вида:

Месяц	День	Тема	Урок	Лабораторная
Декабрь 2017	13	HTML и поиск	ОК	
	20	CSS и Bootstrap		
Январь 2018	3	Переменные и функции		
	10	Структура программы		
	17	Массивы и объекты		
	24	Взаимодействие с веб-страницей		
	31	Взаимодействие с сервером		

10. Помести в раздел **"HTML и поиск"** материалы и ссылки, которые помогли в выполнении этой лабораторной, чтобы обращаться к ним при необходимости в дальнейшем.
11. Размести ссылку на страницу о развитии своего проекта (из пункта 7) на странице **"Об авторе"** (из пункта 1) в разделе **"Текущие проекты"**.
12. Преврати текст ячейки **"HTML и поиск"** в ссылку на раздел **"Лабораторная 1. HTML и поиск"**.
13. Попробуй использовать для оформления цитат из пункта 2 теги `<em>`, `<cite>`, `<i>` и сравни их действие. Выбери наиболее подходящий.
14. Попробуй удалить некоторые из закрывающих тегов и понаблюдай за эффектом.
15. Проверь страницу на соответствие правилам языка HTML.

## Описание действий

### 1. Создай страницу с названием "**Об авторе**".

Для начала необходимо взять за основу готовый шаблон с минимальной структурой. Подходящий вариант находится на портале W3Schools.com в разделе **HTML5 Tutorial**.

HTML-код является обычным текстом, потому для создания веб-страницы достаточно открыть текстовый редактор, скопировать в него текст шаблона и сохранить файл под названием `about.html`.

Некоторые совсем простые редакторы могут самостоятельно менять расширение файла на `about.html.txt`. В этом случае следует поменять расширение файла самостоятельно.

Содержимое тега `<head>` является служебной информацией для браузера и влияет на отображение страницы. Этот элемент браузер обрабатывает в самом начале и уже после начинает отображать содержимое тега `<body>`. Именно здесь располагается тег `<title>`, содержащий название веб-страницы.

Содержимое тега `<body>` является непосредственно содержанием веб-страницы, которое пользователь наблюдает в браузере.

### 2. Размести на странице пару любимых цитат.

Абзацы обрамляются тегами `<p>` и `</p>`. Чтобы перейти на новую строку, не создавая новый абзац, следует использовать тег разрыва строки `<br>` (он является "пустым" и не имеет закрывающего тега).

Так как, по сути, смысловую нагрузку несет именно открывающий тег `<p>` (он создает новый абзац с отступом), некоторые разработчики закрывающий тег `</p>` опускают. Но является ли данный подход правильным?

В случае подобных вопросов стоит обратиться к одному из ресурсов W3C (Консорциума всемирной паутины, действующей главой которого является изобретатель WWW Тим Бернерс-Ли), содержащим наиболее важные сведения об использовании элементов – **W3C Cheat Sheet**.

Поиск по запросу "p" выдаст краткое описание элемента и ссылку на спецификацию. В тексте спецификации содержится целый раздел "Tag omission in text/html". Помимо этого, далее указываются примечания касательно использования тегов и полезные примеры.

В целом, проще и надежнее всегда указывать закрывающий тег.

### 3. Создай на странице раздел "**Коротко обо мне**", помести в него небольшой параграф с информацией о себе.

Для создания заголовка раздела используются теги группы `<h1>` – `<h6>`.

4. Создай раздел **"Увлечения"** и размести в нём нумерованный список с наиболее любимыми занятиями.

Для создания нумерованного списка используются теги `<ol>` и `<li>`.

5. Создай раздел **"Текущие проекты"** и помести в нём ссылку на страницу курса "Основы веб-программирования".

Для создания ссылки используется тег `<a>` и его атрибут `href`. Атрибут помещается в открывающем теге, в качестве значения следует указать адрес ресурса в интернете (URL):

<https://dfisun.github.io/webdev-w2018/>

6. Помести картинку в самом начале страницы. Укажи для неё альтернативный текст с описанием.

Для размещения картинки используется тег `<img>` с атрибутами `src` и `alt`.

7. Создай отдельную страницу о развитии своего проекта. Дай ей подходящий заголовок и название.

Порядок действий аналогичен рассмотренному в пункте 1.

8. Создай раздел **"Дневник разработки"** с подразделами **"План действий"** и **"Лабораторная 1. HTML и поиск"**.

Для создания заголовков используются теги группы `<h1>` – `<h6>`. Для создания подзаголовка нужно перейти на следующий уровень – взять тег с большим числом.

9. Помести в раздел "План действий" таблицу следующего вида:

Месяц	День	Тема	Урок	Лабораторная
Декабрь 2017	13	HTML и поиск	ОК	
	20	CSS, Bootstrap		
Январь 2018	3	Переменные и функции		
	10	Структура программы		
	17	Массивы и объекты		
	24	Взаимодействие с веб-страницей		
	31	Взаимодействие с сервером		

Для создания таблицы используются теги `<table>`, `<tr>`, `<td>`. Для создания заголовочных ячеек вместо тегов `<td>` следует использовать `<th>`.

Так как структура таблицы весьма монотонна, следует активно использовать копирование для ускорения работы.

Для размещения ячейки на нескольких строках следует использовать атрибут `rowspan`.

Чтобы ячейки имели видимые границы, можно задать в теге `<table>` атрибут `border`:

```
<table border="1">
```

Хотя это работающее решение, в дальнейшем стоит предпочесть оформление таблиц с помощью стилей.

10. Помести в раздел **"HTML и поиск"** материалы и ссылки, которые помогли в выполнении этой лабораторной.

Порядок действий аналогичен рассмотренному в пункте 5.

11. Размести ссылку на страницу о развитии своего проекта на странице **"Об авторе"** в разделе **"Текущие проекты"**.

В случае, когда обе страницы располагаются на одном ресурсе (в данном случае — в одной папке или директории), можно использовать локальные ссылки.

В общем случае указывается путь из директории с исходной страницей к странице назначения (включая имя файла). Так как обе страницы располагаются рядом, следует указать имя файла. Например:

```
<a href="webdev-diary.html">Дневник разработки</a>
```

12. Преврати текст ячейки **"HTML и поиск"** в ссылку на раздел **"Лабораторная 1. HTML и поиск"**.

Для создания ссылки на конкретный элемент веб-страницы следует указать атрибут `id` у того элемента, на который мы желаем сослаться. Например:

```
<h2 id="lab1">
```

Далее следует создать обычную ссылку с помощью тега `<a>`, указав в качестве значения атрибута `href` знак решетки (number sign) и название идентификатора (ссылка должна располагаться в том же документе, что и элемент, на который она ссылается). К примеру:

```
<a href="#lab1">HTML и поиск</a>
```

Вероятно, из-за небольшого размера страницы действие ссылки не будет заметно. Для проверки можно разместить сразу после таблицы картинку, вручную увеличив её размеры с помощью атрибутов `width` и `height` так, чтобы раздел **"Лабораторная 1. HTML и поиск"** не был виден при просмотре таблицы. Убедившись, что после нажатия на ссылку страница прокручивается до нужного раздела, удали картинку.

Чтобы сослаться на конкретный элемент другой страницы, нужно указать знак решетки с названием идентификатора после адреса. Например, сослаться на первую лабораторную можно и на странице **"Об авторе"**:

```
<a href="webdev-diary.html#lab1">Дневник разработки – лабораторная 1</a>
```

13. Попробуй использовать для оформления цитат из пункта 2 теги `<em>`, `<cite>`, `<i>` и сравни их действие. Выбери наиболее подходящий.

При использовании любого тега следует уделять особое внимание его семантическому значению (проще говоря — назначению), так как помимо традиционных браузеров существуют браузеры с текстовым интерфейсом, а также специализированные приложения для людей с ограниченными возможностями.

В случае, когда разработчик ориентируется исключительно на внешний вид страницы и не заботится о семантике используемых элементов, страница может стать совершенно недоступной для некоторых пользователей.

Для уточнения значения элемента следует желательнее обращаться к ресурсу **W3C Cheat Sheet**, содержащему также рекомендации по использованию тега и ссылку на спецификацию, или в иной авторитетный справочник HTML (например, **W3Schools HTML Tag Reference**).

В случае оформления цитаты уместно использование тега `<cite>`. Тег `<em>` используется с целью обозначить важность содержимого или поставить смысловое ударение. Тег `<i>` ранее использовался для оформления наклонного текста и на данный момент является устаревшим.

Следует воздерживаться от использования устаревших элементов! С момента введения стандарта HTML5 для визуального оформления используются каскадные таблицы стилей (CSS). Их мы рассмотрим на следующем занятии.

14. Попробуй удалить некоторые из закрывающих тегов и понаблюдай за эффектом. Верни страницу к исходному виду.

В случае, если содержание веб-страницы не удовлетворяет правилам, браузер в автоматическом режиме добавляет необходимые элементы. Зачастую полученный результат является неправильным с точки зрения разработчика, но позволяет браузеру отобразить страницу.

Чтобы увидеть, какие изменения произвел браузер, следует щелкнуть на интересующем элементе правой кнопкой мыши и выбрать "Inspect Element" или открыть Инструменты разработчика:

Firefox: Tools > Web Developer > Inspector

Chrome: More Tools > Developer Tools > Elements

**NB!** Выбор действия "View Page Source" выдаст исходное содержимое веб-страницы до внесения изменений со стороны браузера.

15. Проверь страницу на соответствие правилам языка HTML.

Консорциум всемирной паутины предлагает специальный ресурс для проверки документов различных Web-форматов на соответствие правилам используемого языка — **W3C Markup Validator**.

Ресурс принимает веб-адрес, файл или исходный текст документа. При наличии ошибок сообщается их местонахождение, характеристики и рекомендации по исправлению.

Соответствие правилам языка — один из признаков хорошей веб-страницы.

## Рекомендуемые материалы

1. W3Schools HTML5 Tutorial  
<https://www.w3schools.com/html/>
2. W3Schools HTML Tag Reference  
<https://www.w3schools.com/tags/>
3. W3C Cheat Sheet  
<https://www.w3.org/2009/cheatsheet/>
4. W3C Markup Validator  
<http://validator.w3.org/>

## Дополнительные материалы

1. W3C Web Accessibility Initiative: Easy Checks - A First Review of Web Accessibility  
<https://www.w3.org/WAI/eval/preliminary>

## Лицензия



Это произведение доступно по лицензии Creative Commons С указанием авторства-С сохранением условий 4.0 Всемирная (CC BY-SA 4.0).

Эта лицензия разрешает свободно распространять данное произведение и создавать на его основе производные при соблюдении следующих условий:

- указание авторства — Вы должны обеспечить соответствующее указание авторства, предоставить ссылку на лицензию, и обозначить изменения, если таковые были сделаны
- на тех же условиях — если Вы преобразовываете данное произведение или создаете на его основе производное, Вы должны распространять использованные фрагменты произведения только на условиях этой или совместимой лицензии.

Это краткое резюме (не имеющее юридической силы) полного текста лицензии, доступного по адресу <https://creativecommons.org/licenses/by-sa/4.0/>.