

## Практическое занятие №2. Формы и таблицы

### Задание №1

Проверьте запущен ли у Вас Open Server. Папка с материалами урока (html2.loc) должна быть размещена в директории /domains сервера.

Для этого Вам нужно:

1. Перенести архив в директорию ...\\OpenServer\\domains. Распаковать архив в неё.
2. Запустить программу Open Server x64.exe
3. И запустить сам сервер (флажок должен стать зеленым).
4. Если сервер был запущен с самого начала, то его нужно перезапустить.
5. Зайти в браузер Chrome.
6. В новой вкладке , перейти по адресу <http://html2.loc/>.
7. Убедитесь, что всё работает.
8. Нажмите на кнопку "Начнем".
9. Далее, кликните на надпись: "Задание №2".
10. Перед вами будет пустая страница, на которой Вы будете видеть результат своих действий.

### Задание №2

Таблицы важный элемент HTML страниц. Ранее они очень активно использовались в качестве инструмента структурирования страницы. Сейчас таблицы используются только по назначению – для наглядного отображения информации.

Вам нужно навестать каркас таблицы как показано на рисунке ниже. Элемент `<table>` определяет саму таблицу. Элементы `<thead>`, `<tfoot>`, `<tbody>` определяют соответственно верхнюю часть таблицы (шапку), нижнюю часть таблицы и тело таблицы. В принципе элементы `<thead>`, `<tbody>` и `<tfoot>` необязательны, однако рекомендуется их использовать для более ясной структуры таблицы в коде.

```
<table>
  <thead></thead>
  <tfoot></tfoot>
  <tbody></tbody>
</table>
```

Элемент `<tr>` (table row) объявляет строку таблицы. Элемент `<td>` объявляет ячейку таблицы. ВНИМАНИЕ! Элемент `<td>` может находиться ТОЛЬКО внутри элемента `<tr>`. Добавляем к нашей таблице строки и ячейки как показано на картинке снизу.

```
<table>
  <thead>
    <tr>
      <td>Заголовок 1</td>
      <td>Заголовок 2</td>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td>Ячейка 1, TFOOT</td>
      <td>Ячейка 2, TFOOT</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Ячейка 1, TBODY</td>
      <td>Ячейка 2, TBODY</td>
    </tr>
  </tbody>
</table>
```

Тег `<th>` является аналогом тега `<td>`, он используется для того что бы создать ячейку-заголовок. В секции `<thead>` замените все теги `<td>` на `<th>`. Понаблюдайте за изменением поведения таблицы.

```
<thead>
  <tr>
    <th>Заголовок 1</th>
    <th>Заголовок 2</th>
  </tr>
</thead>
```

### Задание №3

Создаём таблицу как на рисунке ниже.

```
<h1>Дневная выработка</h1>
<table>
  <thead>
    <tr>
      <th>Имя сотрудника</th>
      <th>Количество</th>
      <th>Стоимость</th>
      <th>Сумма</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Jonn</td>
      <td>5</td>
      <td>20</td>
      <td>100</td>
    </tr>
    <tr>
      <td>Nick</td>
      <td>7</td>
      <td>20</td>
      <td>140</td>
    </tr>
  </tbody>
</table>
```

Далее добавляем секцию <tfoot> как показано на изображении ниже.

```
<tfoot>
  <tr>
    <td>Всего</td>
    <td></td>
    <td></td>
    <td>240</td>
  </tr>
</tfoot>
```

Сейчас мы столкнулись с дилеммой, в строке внутри <tfoot> нам нужно всего две ячейки, а в таблице 4 столбца. Для этого существует атрибут слияния ячеек **colspan="n"**, где n – это количество ячеек, которые должны быть объединены по горизонтали. Используя атрибут **colspan** производим слияние ячеек.

```
<tfoot>
  <tr>
    <td colspan="3" >Всего</td>
    <!--
    <td></td>
    <td></td>
    -->
    <td>240</td>
  </tr>
</tfoot>
```

#### Задание №4

Используя полученные навыки создаем таблицу как на рисунке снизу.

1	1
1 + 1	
1	1
1 + 1	

#### Задание №5

Иногда возникает необходимость сливать ячейки таблицы не горизонтально, а вертикально для этого используем атрибут **rowspan**. Наберите код на картинке ниже и посмотрите, что получится.

```
<table>
  <tr>
    <td>1</td>
    <td rowspan="2">2</td>
  </tr>
  <tr>
    <td>1</td>
  </tr>
</table>
```

### Задание №6

Закрепляем слияние ячеек, верстаем табличку как на рисунке ниже.

2	1	2	
	1	1	1

### Задание №7

Web Формы служат для передачи данных от пользователей сайта на сервер. В этом задании мы шаг за шагом создадим простую web форму.

Тег `<form>` создаёт форму на web странице. Создайте тег формы в файле задания.

```
<form>
</form>
```

Атрибут **METHOD** тега `<form>` определяют метод запроса формы. Значения этого атрибута могут быть **GET** и **POST**.

**GET** – используется для передачи небольшого объема простых данных. Данные передаются в самой адресной строке запроса в виде пар ключ = “значение”, которые присоединяются в этом случае к адресу после вопросительного знака и разделяются между собой амперсандом (символ &). Удобство использования метода GET заключается в том, что адрес со всеми параметрами можно использовать неоднократно, сохранив его, например, в закладки браузера, а также менять значения параметров прямо в адресной строке. Данные передаются в открытом виде.

**POST** – используется для передачи любого объема данных, включая передачу файлов. Данные POST гораздо передаются в более скрытом варианте чем GET (однако это не значит, что их не сможет перехватить злоумышленник, для систем с высокими требованиями безопасности используют другие методы защиты информации) поэтому их предпочтительнее использовать для авторизации пользователей на сайте и передачи другой пользовательской информации.

*Передача пользовательской информации через GET может привести к неожиданным последствиям, например, пользователь решил поделиться ссылкой с другом или запостить ссылку на ваш ресурс у себя на страничке в соцсети и взял ссылку из строки браузера. Однако в строке браузера могут содержаться данные, отправленные GET запросом и пользователь может предоставить другим информацию, которой вовсе не стоило делиться.*

Прописываем нашей форме метод передачи GET

```
<form method="GET" >  
</form>
```

Атрибут **ENCTYPE** определяет способ кодирования данных формы при их отправке на сервер. Он может принимать следующие значения

application/x-www-form-urlencoded (по умолчанию) вместо пробелов ставится +, символы вроде русских букв кодируются их шестнадцатеричными значениями (например, %D0%90%D0%BD%D1%8F вместо Аня).

multipart/form-data - данные не кодируются. Это значение применяется при отправке файлов.

text/plain - Пробелы заменяются знаком +, буквы и другие символы не кодируются.

Добавляем нашей форме атрибут enctype как показано на картинке ниже.

```
<form method="GET" enctype="application/x-www-form-urlencoded" >  
</form>
```

Атрибут **ACTION** указывает обработчик, к которому обращаются данные формы при их отправке на сервер. Если атрибут action отсутствует, текущая страница перезагружается, возвращая все элементы формы к их значениям по умолчанию.

Устанавливаем атрибут action как показано на изображении ниже.

```
<form action="/task-7/" method="GET" enctype="application/x-www-form-urlencoded" >
</form>
```

На данный момент у нас на странице есть форма, которая передаёт данные методом GET на страницу <http://html2.loc/task-7/> . Но сейчас она пустая.

Тег **<input>** позволяет создавать различные элементы формы. С помощью **<input>** можно создавать текстовые поля, различные кнопки, переключатели и флажки. В примере ниже мы научимся использовать **<input>** в качестве текстового поля и кнопки. Создаём текстовые поля как показано ниже.

```
<form action="/task-7/" method="GET" enctype="application/x-www-form-urlencoded" >
  <input type="text" />
  <input type="text" />
  <input type="submit" />
</form>
```

Как Вы можете видеть атрибут type определил внешний вид и функциональность элементов input.

**type="text"** – создает текстовое поля для ввода.

**type="submit"** – создал кнопку, при нажатии на которую происходит отправка формы по указанному адресу. В нашем случае это текущая страница с формой.

Если сейчас Вы заполните поля формы произвольными значениям, то при отправке формы вы увидите, что к URL станицы добавился "?".

Это означает что форма была отправлена, но никаких данных в ней не было. Для того что данные с формы начали поступать на сервер, необходимо добавить полям форты атрибут **name**, который определяют название пересылаемого параметра.

Добавьте атрибуты name как показано на изображении ниже.

```
<form action="/task-7/" method="GET" enctype="application/x-www-form-urlencoded" >
  <input type="text" name="name" />
  <input type="text" name="family" />
  <input type="submit" />
</form>
```

Теперь попробуйте заполнить и отправить форму снова, понаблюдайте за изменениями.

### Задание №8

В этом задании Вам нужно создать форму, которая будет отсылать данные на страницу задания методом GET. Поля формы, которые нужно реализовать:

Назначение поля	Тип поля (type)	Название поля(name)	Пример заполнения
День рождения	text	day	03
Месяц рождения	text	month	01
Год рождения	text	year	1999
Кнопка отправить	submit		

Отправьте свою форму и убедитесь в ее работоспособности.

### Задание №9

Атрибут **VALUE** позволяет указать предустановленное значение полям формы.

Атрибут **PLACEHOLDER** позволяет указать подсказку, которая будет появляться если поле формы пустое.

Сейчас попробуем применить эти атрибуты и посмотреть на особенности их поведения.

Верстаем форму как показано на изображении ниже.

```
<form action="/task-9/" method="GET" >
  <input type="text" name="search" value="Поиск" placeholder="Поиск по сайту" >
</form>
```

Исследуйте поведение поля, когда оно оказывается пустым и заполненным текстом.

### Задание №10

В этом задании мы создаём простейшую форму авторизации. Для этого мы используем метод POST и скрываем вводимое значение пароля через атрибут type с значением "password".

```
<form action="/task-10/" method="POST" >
  <input type="text" name="username" placeholder="Ваш логин" />
  <input type="password" name="password" placeholder="Ваш пароль" />
  <input type="submit" value="Авторизоваться на сайте" />
</form>
```

После того как Вы наверстали форму попробуйте авторизоваться, используя логин: admin, пароль: 123456.

### Задание №11

Для ввода комментариев, отзывов и сообщений в формах предусмотрен элемент **textarea**. Размеры этого поля могут определяться атрибутами:

**cols** – определяет ширину поля в количестве символов.

**rows** – определяет высоту поля в количестве строк.



Создадим форму с одним таким полем, как показано на рисунке ниже.

```
<form action="/">
  <textarea cols="30" rows="10" name="text" placeholder="Комментарий" ></textarea>
</form>
```

У этого элемента есть особенность, которую мы сейчас исследуем. Значение этого поля определяется не атрибутом value, а текстом внутри тега. Добавьте любой текст внутри тега `<textarea>` и посмотрите на результат.

### Задание №12

Используя полученные знания навестайте форму отправки комментария на сервер. Поля формы представлены в таблице ниже.

Метод отправки формы		GET	
Адрес обработчика формы		http://html2.loc/task-12/	
Поле	Name	Элемент	Placeholder
Имя пользователя	username	input type='text'	Введите Ваше имя
Фамилия пользователя	family_name	input type='text'	Введите Вашу фамилию
Текст комментария	comment	textarea	Комментарий
Кнопка отправки формы		input type="submit"	

Убедитесь, что Ваша форма отправляет данные на сервер.

### Задание №13

Элемент формы `<select>` позволяет нам создавать выпадающие списки. Элементы `<option>` всегда размещаются внутри `<select>` и определяют отдельные строки выпадающего списка. Создаём простую форму с одним выпадающим списком, как на картинке ниже.

```
<form action="/task-13/" >
  <p>Выберите вариант доставки:</p>
  <select name="delivery">
    <option value="1" >Самовывоз</option>
    <option value="2" >Доставка до точки выдачи</option>
    <option value="3" >Адресная доставка</option>
  </select>
  <input type="submit" />
</form>
```

Обратите внимание, что на сервер приходит значение атрибута value выбранного элемента `<option>`.

## Задание №14

Чекбоксы и радиокнопки являются инструментом выбора из определённого списка вариантов. Отличие между этими элементами состоит в том, что радиокнопка позволяет выбрать только одно значение из предложенных, а чекбоксы - несколько.

Создадим форму в которой протестируем эти элементы. Начнем с радиокнопок. Наверстайте форму, как показано на рисунке ниже. Для того что у нас получилось сверстать данную форму нам необходим тег `<label>`. Тег `<label>` устанавливает связь между текстом (или другим HTML элементом) и элементом формы.

```
<form action="/task-14/">
  <label>
    <input type="radio" name="radiobutton" value="1" />
    Значение №1
  </label>
  <label>
    <input type="radio" name="radiobutton" value="2" />
    Значение №2
  </label>
  <label>
    <input type="radio" name="radiobutton" value="3" />
    Значение №3
  </label>

  <input type="submit" />

</form>
```

Обратите внимание на следующие моменты:

1. Одновременно возможно выбрать только одну из радиокнопок.
2. У всех радиокнопок одной группы атрибут `name` должен быть одинаковым
3. При клике по тексту в теге `<label>` происходит активация советующего элемента формы.
4. На сервер отправляется значение.

Также тег `<label>` можно использовать следующим образом (поскольку включение элемента в `label` не всегда будет являться хорошим вариантом с точки зрения стилизации):

```
<form>
  <input id="answer-1" type="radio" name="answer" value="1">
  <label for="answer-1">Вариант 1</label><br>
  <input id="answer-2" type="radio" name="answer" value="2">
  <label for="answer-2">Вариант 2</label><br>
  <input id="answer-3" type="radio" name="answer" value="3">
  <label for="answer-3">Вариант 3</label><br>
  <input id="answer-4" type="radio" name="answer" value="4">
  <label for="answer-4">Вариант 4</label><br>
  <input id="answer-5" type="radio" name="answer" value="5">
  <label for="answer-5">Вариант 5</label><br>
</form>
```

Такая форма записи более гибкая и позволит расположить `label` за пределами формы или сделать несколько `label` для одного элемента.

Поскольку id должен быть уникальным то в этом случае можно воспользоваться следующим правилом {имя поля}\_{значение}. Также для большей уникальности можно добавить в начале имя (name, id или часть action) от формы {ид формы}\_{имя поля}\_{значение}.

### Задание №15

Верстаем форму с чекбоксами как на изображении ниже.

```
<form action="/task-15/">
  <label>
    <input type="checkbox" name="checkbox" value="1" />
    Значение №1
  </label>
  <label>
    <input type="checkbox" name="checkbox" value="2" />
    Значение №2
  </label>
  <label>
    <input type="checkbox" name="checkbox" value="3" />
    Значение №3
  </label>

  <input type="submit" />

</form>
```

Обратите внимание на следующие моменты:

1. Одновременно возможно выбрать несколько чекбоксов.
2. Если не выбран ни один чекбокс то форма не посылает никакого значения. Просто отправляется пустая форма.
3. У всех чекбоксов одной группы атрибут name должен быть одинаковым

Вы можете заметить еще один нюанс работы чекбоксов – в текущей форме мы можем выбрать несколько чекбоксов, но на сервер поступает только последнее выбранное значение. Для того чтобы получить все выбранные значения нужно воспользоваться «хаком». Нужно к атрибуту name добавить «[]» в конце и тем самым показать, что это массив значений.

```
name="checkbox[]"
```

Добавьте к атрибутам name «[]» и протестируйте форму снова.

### Задание №16

Для формы, которая передаёт файлы, не забываем указать метод передачи POST и enctype. За загрузку файлов отвечает элемент формы `<input type="file">`. Верстаем форму как показано на

изображении ниже и пытаемся загрузить файл изображение **jpg** формата, используя данную форму.

```
<form action="/task-16/" method="POST" enctype="multipart/form-data" >

  <input type="file" name="img" />
  <input type="submit" />

</form>
```

Посмотрите также атрибут `accept` <http://htmlbook.ru/html/input/accept> для `input type='file'`, который можно использовать для уточнения типа поддерживаемого файла, для того чтобы упростить пользователю его поиск при выборе.

### Задание №17

Иногда возникает необходимость передать данные в форме скрыто от пользователя. Это может быть техническая информация для сценария обработчика для этого используется уже знакомый нам тег `<input type="hidden">`.

Часто нужно что бы поля формы были обязательно заполненными. Сделать это очень просто. Нужно добавить атрибут **required** к полю формы что бы он стал обязательным для заполнения.

Верстаем форму, как на изображении ниже, чтобы увидеть, как это работает.

```
<form action="/task-17/" >
  <input type="hidden" name="top-secret" value="1248" />
  <input type="text" name="code" required />
  <input type="submit" />
</form>
```

В поведении этой формы можно заметить следующее:

1. Невозможно отправить форму если не заполнен параметр `code`.
2. Данные со скрытого поля отправляются на сервер.

### Домашнее задание №1

Форматы изображений в Web. Узнайте больше прочитав [статью с внешнего ресурса](#).

### Домашнее задание №2

Вам нужно выбрать трёх своих любимых персонажей из фильмов, сериалов или книг и навестать таблицу-анкету, в которой каждый персонаж будет иметь свою колонку. В этой таблице должна быть фотография каждого персонажа, имя, вид деятельности, ссылка на его сайт или страничку и другие характеристики на Ваш выбор.

### Домашнее задание №3

Создайте форму регистрации со следующими полями.

Название поля	Описание поля
Имя пользователя	Короткое текстовое поле
Логин	Короткое текстовое поле
Пароль	Поле для ввода пароля
Подтверждение пароля	Поле для ввода пароля
Фото профиля	Поле загрузки файла
Тип профиля	Выпадающий список. Варианты: обычный, премиум
Пол	Радиокнопка (мужской, женский)
Рассылка новостей по темам	Чекбоксы. Темы рассылки: HTML5, CSS, SEO, PHP7
О себе	Многострочное текстовое поле