

## Практическая работа №1

### «Наполнение и оформление веб-страниц»

**Цель работы:** познакомиться с основами синтаксиса и способами использования языка разметки гипертекстовых документов HTML 5 и языка оформления CSS 3.

#### ЗАДАНИЯ

##### А. Подготовка к работе:

1. Запустите (в случае необходимости – установите) менеджер виртуальных машин VMware Player на **реальном** компьютере.
2. В программе VMware Player запустите образ **виртуального** компьютера **WWW** с предустановленным ПО: Apache (HTTP-сервер), PHP (интерпретатор) и MySQL (СУБД).
3. В адресной строке браузера с реального компьютера откройте поочередно ссылки <http://www/>, <http://www/info.php> и <http://www/sqltest.php>. Убедитесь в работоспособности перечисленных выше программных компонентов виртуального веб-сервера.
4. В проводнике на реальном компьютере введите адрес [\\www](http://www), укажите по запросу имя пользователя – **Администратор**; пароль – **1234**. Отобразится список общих папок веб-сервера, в т.ч. папка **www** – **корневая папка сайта** (локальный путь – C:\wamp\www).

Если проводник сообщает, что нет доступа к ресурсу, то на реальном компьютере (для ОС Windows Vista/ 7 /8.x/10) откройте «Панель управления» → «Сеть и Интернет» → «Центр управления сетями и общим доступом» → «Изменить дополнительные параметры общего доступа». В нужном профиле выберите «Включить сетевое обнаружение» и «Включить общий доступ, чтобы сетевые пользователи могли читать и записывать файлы в общих папках». Сохраните изменения и повторите попытку.

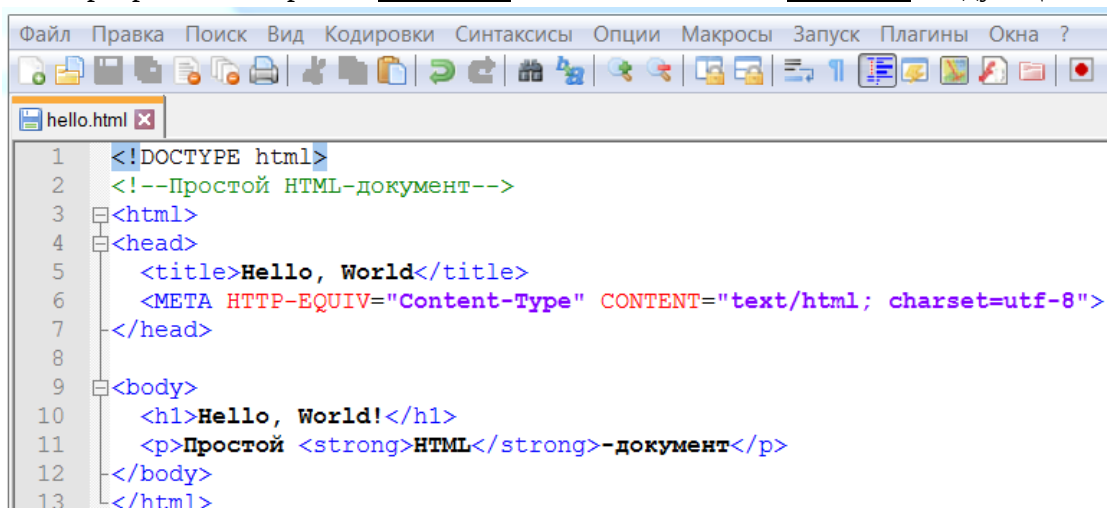
Возможной причиной блокирования также может быть **пустой пароль** у текущего пользователя реального компьютера.

*Примечание.* если запрещено изменение настроек на реальном компьютере, редактирование файлов сайта в дальнейшем возможно прямо на виртуальном компьютере.

5. Запустите на реальном компьютере специализированный текстовый редактор Notepad++ (из предложенной папки ПО). Возможно использование любого другого текстового редактора (без визуальных средств разработки).

##### Б. Язык разметки гипертекста HTML

6. В программе Notepad++ включите синтаксис HTML и наберите следующий HTML-код:



```

1  <!DOCTYPE html>
2  <!--Простой HTML-документ-->
3  <html>
4  <head>
5      <title>Hello, World</title>
6      <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
7  </head>
8
9  <body>
10     <h1>Hello, World!</h1>
11     <p>Простой <strong>HTML</strong>-документ</p>
12 </body>
13 </html>
  
```

Сохраните его под именем hello.html в кодировке UTF-8 в корневую папку сайта.

Откройте отображение документа в браузере с помощью меню «Запуск».

Еще раз откройте документ, но уже без помощи Notepad++ в браузере с реального компьютера, введя адрес <http://www/hello.html>.

### Теги.

В приведенном коде слова, взятые в угловые скобки < и >, – это теги HTML, инструкции браузеру, который *интерпретирует* тот или иной фрагмент HTML-кода в определенный элемент Web-страницы: абзац, заголовок или текст.

Так, теги <h1> и </h1> превращают фрагмент текста, находящийся между ними, в заголовок. Тег <h1> помечает начало фрагмента, на который распространяется действие тега, и называется *открывающим*. Тег </h1> называется *закрывающим*. Сам фрагмент, заключенный между открывающим и закрывающим тегами, – *содержимое* тега. К содержимому и применяется действие тега.

Теги <h1> и </h1> считаются одним тегом <h1>. Такой тег называется *парным*. Парный тег <P> создает абзац; содержимое тега станет текстом этого абзаца. Абзац будет отображаться с отступами сверху и снизу.

В теге <meta> нет ни содержимого, ни закрывающей пары – это *одинарный* тег.

Параметры Web-страницы, не отображаемые на экране и предназначенные для Web-обозревателя, называют *метаданными*. Это данные, описывающие другие данные, в нашем случае – Web-страницу. HTML-теги, задающие метаданные – *метатеги*.

Тег <meta> – пример метатега, задающего в том числе кодировку UTF-8, в которой сохранена Web-страница. Существуют аналогичные теги, задающие кодировки Win-1251, КОИ-8, кодировка западноевропейских и восточноазиатских языков и др.

Другой одиночный метатег <!DOCTYPE> в самом начале HTML-кода указывает браузеру версию языка разметки в документе (в нашем случае HTML 5).

### Атрибуты тегов.

В теге <META> присутствуют *атрибуты* тега – параметры, уточняющие его действие.

Каждый атрибут тега имеет *имя*, за которым ставится знак равенства, и *значение* данного атрибута, взятое в двойные кавычки. Так, в примере атрибут HTTP-EQUIV имеет значение "Content-Type" и задает тип документа. Атрибут CONTENT имеет значение "text/html; charset=utf-8", обозначающее, что данный документ – текст на языке HTML, и что он набран в кодировке UTF-8.

Атрибуты тегов бывают *обязательными* и *необязательными*. Необязательные атрибуты могут не указываться; в таком случае тег ведет себя соответственно значению этого атрибута *по умолчанию*. Для каждого HTML-тега существует свой набор обязательных и необязательных атрибутов.

7. Откройте справочник по HTML «[htmlbook.ru](http://htmlbook.ru) | Для тех, кто делает сайты» по адресу <http://htmlbook.ru/>. Найдите в нём описание всех возможных атрибутов тега <P>.

### Вложенность тегов.

В приведенном коде видно, что одни теги *вложены* в другие. Так, тег <strong> *вложен* в тег <P>, являясь частью его содержимого. Тег <P>, в свою очередь, вложен в тег <body>, а тот – в тег <html>.

Когда Web-обозреватель встречает тег, вложенный в другой тег, он накладывает действие "внутреннего" тега на эффект "внешнего". Так, действие тега `<strong>` будет наложено на действие тега `<P>`, и фрагмент абзаца окажется выделенным полужирным шрифтом, при этом оставаясь частью этого абзаца.

Важно: при вложении порядок следования закрывающих тегов должен быть обратным тому, в котором следуют теги открывающие. Теги со всем их содержимым должны полностью вкладываться в другие теги, не оставляя "хвостов" снаружи. Иначе браузер отобразит страницу неправильно.

Тег, в который непосредственно вложен данный тег, называется *родительским*, или *родителем*. В свою очередь, тег, вложенный в данный тег, – *дочерний*, или *потомок*.

Уровень вложенности тегов в коде HTML обозначают с помощью отступов, которые ставят слева от соответствующего тега и создают с помощью пробелов.

### **Блочные и встроенные объекты HTML-страницы.**

Парный тег `<P>` в языке HTML используется для создания абзаца.

Абзац – это пример *независимого* элемента Web-страницы, который отображается отдельно от других элементов. Такие элементы Web-страниц называются *блочными*, или *блоками*. Блочными объектами являются все виды заголовков, списки, таблицы, контейнеры `<div>` и др.

Тег `<strong>` – это нечто большее, чем просто выделение текста. Он даёт фрагменту текста, являющемуся его содержимым, *особое значение* с точки зрения и Web-обозревателя, и поискового робота.

Элемент Web-страницы, который этот тег создаёт, не является независимым и не отображается отдельно от "соседей", а *принадлежит* блочному объекту – абзацу. Такие элементы называются *встроенными*. Встроенными объектами являются гиперссылки, а также внедряемые внешние объекты – рисунки, звук и видео (см. следующий пример).

### **Секции страницы.**

В данном примере применены теги, которые делят Web-страницу на две *секции*.

В парном теге `<head>` находится секция *заголовка* Web-страницы (не путать с заголовком-абзацем, который создается с помощью тега `<H1>`!). В эту секцию помещают сведения о параметрах Web-страницы, не отображаемые на экране и предназначенные исключительно для обозревателя.

Секция *тела* Web-страницы создается парным тегом `<body>`. В теле – само содержимое Web-страницы, то, что будет выведено на экран.

И заголовок, и тело Web-страницы находятся внутри парного тега `<html>`, который расположен на самом высшем (нулевом) уровне вложенности и не имеет родителя.

Любая Web-страница должна быть правильно отформатирована – иметь секции заголовка и тела. Только тогда она будет считаться корректной с точки зрения стандарта HTML 5.

Вообще, в сети Интернет можно найти специализированные сайты-валидаторы, позволяющие проверить код ваших веб-страниц на соответствие утвержденным Консорциумом World Wide Web стандартам HTML 5 и CSS 3 (см., например, <https://github.com/w3c/validator-suite>).

8. Наберите код следующего примера:

```

1 <!DOCTYPE html>
2 <!--Теги HTML-->
3 <html>
4 <head>
5   <title>Теги HTML</title>
6   <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
7 </head>
8 <body>
9   <h1>Теги HTML</h1>
10
11   <h3>Нумерованный список</h3>
12   <ol>
13     <li>Во-первых</li>
14     <li>Во-вторых</li>
15     <li>В-третьих</li>
16   </ol>
17   <hr>
18   <h3>Маркированный список</h3>
19   <ul>
20     <li>Такой</li>
21     <li>Другой</li>
22     <li>Совсем другой</li>
23   </ul>
24   <hr>
25   <h3>Гиперссылка</h3>
26   <p>
27     <a href="http://apod.nasa.gov/apod/calendar/allyears.html" target="_blank">
28       Астрономическая картинка на каждый день от НАСА</a>
29   </p>
30   <hr>
31   <h3>GIF-изображение</h3>
32   <p>
35   </p>
36 </body>
37 </html>

```

Сохраните его в файле more.html в папке сайта. Туда же скопируйте файл silly.gif с изображением в формате GIF.

Откройте страницу в обозревателе.

### Таблицы.

*Таблица* – блочный элемент Web-страницы. Это значит, что она размещается отдельно от всех остальных блочных элементов: абзацев, заголовков, больших цитат, аудио- и видеороликов.

Таблицы HTML создаются в четыре этапа. На первом этапе с помощью парного тега <table> формируют саму таблицу.

На втором этапе формируют строки таблицы парными тегами <tr>, каждый такой тег создает отдельную строку. Теги <tr> помещают внутрь тега <table>.

На третьем этапе создают ячейки парными тегами <TD> и <TH>. Тег <TD> создает обычную ячейку, тег <TH> – ячейку заголовка, "шапку" соответствующего столбца таблицы. Теги <TD> и <TH> помещают в нужные теги <TR>, создающие строки таблицы.

На четвертом, последнем, этапе указывают содержимое ячеек, которое помещают в соответствующие теги <TD> и <TH>.

9. Наберите код следующего примера:

```

1  <!DOCTYPE html>
2  <!--Таблицы в HTML--><html>
3
4  <head>
5      <title>Таблицы</title>
6      <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
7  </head>
8
9  <body>
10     <h1>Таблицы<hr></h1>
11
12     <h2>Основная таблица</h2>
13
14     <table>
15     <tr>
16         <th></th>
17         <th>Понед-к</th>
18         <th>Вторник</th>
19         <th>Среда</th>
20         <th>Четверг</th>
21         <th>Пятница</th>
22     </tr>
23
24     <tr>
25         <th>1 неделя</th>
26         <td>Математика</td>
27         <td>Химия</td>
28         <td>Математика</td>
29         <td>История</td>
30         <td>Музыка</td>
31     </tr>
32
33     <tr>
34         <th>2 неделя</th>
35         <td>ИЗО</td>
36         <td>Ин. яз. </td>
37         <td>Биология</td>
38         <td>Ин. яз. </td>
39         <td>Физика</td>
40     </tr>
41     </table>
42
43     <hr>
44
45     <h2>Таблица с объединением ячеек</h2>
46     <table>
47
48     <tr>
49         <th></th>
50         <th>Понед-к</th>
51         <th>Вторник</th>
52         <th>Среда</th>
53         <th>Четверг</th>
54         <th>Пятница</th>
55     </tr>
56
57     <tr>
58         <th>1 неделя</th>
59         <td>One</td>
60         <td colspan = 2>Two</td>
61         <td>Three</td>
62         <td rowspan = 2>Four</td>
63     </tr>
64

```

```

65
66 <tr>
67   <th>2 неделя</th>
68   <td>A</td>
69   <td>B</td>
70   <td>C</td>
71   <td>D</td>
72 </tr>
73
74 </table>
75
76 </body>
77 </html>

```

Сохраните код в файле под именем table.html. Проверьте код в браузере.

Во второй таблице примера выполнено объединение некоторых ячеек. Для этого в тегах <TD> и <TH> использованы два необязательных атрибута: 1) COLSPAN – объединяет ячейки по горизонтали, 2) ROWSPAN – по вертикали.

Чтобы объединить несколько ячеек *по горизонтали* в одну:

- Найти в коде HTML тег <TD> (<TH>), соответствующий первой из объединяемых ячеек (если считать ячейки слева направо).
- Вписать в него атрибут COLSPAN и присвоить ему количество объединяемых ячеек, считая и самую первую из них.
- Удалить теги <TD> (<TH>), создающие остальные объединяемые ячейки этой строки.

Объединить ячейки *по вертикали* сложнее:

- 1. Найти в коде HTML строку (тег <TR>), в которой находится первая из объединяемых ячеек (если считать строки сверху вниз).
- 2. Найти в коде этой строки тег <TD> (<TH>), соответствующий первой из объединяемых ячеек.
- 3. Вписать в него атрибут ROWSPAN и присвоить ему количество объединяемых ячеек, считая и самую первую из них.
- 4. В последующих строках удалить теги <TD> (<TH>), создающие остальные объединяемые ячейки.

## В. Язык оформления CSS 3.

### Стили.

Для оформления Web-страниц предназначена технология *каскадных таблиц стилей* (Cascading Style Sheets, CSS). Таблица стилей содержит набор *стилей*, описывающих оформление самой Web-страницы и отдельных ее фрагментов. Стили определяют цвет текста и выравнивание абзаца, отступы между графическим изображением и обтекающим его текстом, параметры рамки таблиц, цвет фона Web-страницы, вид курсора мыши и многое другое.

Каждый стиль должен быть *привязан* к соответствующему элементу Web-страницы (или к самой Web-странице). Привязка может быть *явная*, когда точно указано, какой стиль к какому элементу страницы привязан, или *неявная*, когда стиль автоматически привязывается ко всем элементам страницы, созданным с помощью определенного тега.

Таблица стилей может храниться прямо в HTML-коде Web-страницы или в отдельном файле с расширением .css. Последний подход наиболее соответствует концепции Web

2.0. Концепция Web 2.0 требует, чтобы наполнение (HTML-код), оформление (CSS-код) и активное содержимое (клиентские скрипты) Web-страницы были разделены.

Отдельные стили можно также поместить прямо в тег HTML, создающий элемент Web-страницы; такой подход используется редко и, в основном, при отладке.

Типы стилей:

- а) стили переопределения тега;
- б) стилевые классы;
- в) именованные стили.

10. Наберите следующий код:

```

1  <!DOCTYPE html>
2  <!--Стили в самом HTML-документе-->
3  <html>
4  <head>
5      <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
6      <title>Встроенные стили</title>
7
8      <style type = "text/css">
9          h1 {
10             text-align:center;
11             color:green;
12             border-color:red;
13             border-style:double;
14             border-size: 3px
15          }
16
17          p {
18             background-color: yellow;
19             font-family: monospace
20          }
21
22          .cursive {
23             background-color: yellow;
24             font-family: cursive
25          }
26
27      </style>
28
29  </head>
30
31  <body>
32      <h1>Заголовок H1 в своём стиле</h1>
33      <p>А вот абзац уже в своём стиле</p>
34      <p class = cursive>
35          Абзац в стилевом классе cursive
36      </p>
37  </body>
38  </html>

```

Сохраните его в файле `pagestyle.html`, откройте в браузере.

В примере предложены *встроенные в HTML-код* страницы три стиля. Стили `h1` и `p` – это *стили переопределения* соответствующих тегов. Третий стиль, название которого в описании начинается с точки, – *стилевой класс*.

Первые два стиля применяются ко всем объектам страницы, сформированным тегами `h1` и `p`. Стилиевой класс применится к тем объектам, к которым он явно привязан атрибутом тега `class` (при вызове имя стиля указывается без начальной точки).



*Именованный стиль* аналогичен стилевому классу с теми отличиями, что 1) имя стиля в описании начинается не с точки, а с символа #; 2) именованный стиль привязывается к объекту атрибутом id, а не class. ID – идентификатор объекта страницы для программных скриптов. У каждого объекта должен быть уникальный id, поэтому именованный стиль обычно привязывается к единственному объекту на странице.

Описание стиля – набор *атрибутов стиля* (не путать с атрибутами тега!) в соответствии со стандартом языка CSS.

11. В электронном справочнике найдите описание атрибутов, использованных в стилях h1, p и cursive.

12. Вернитесь в файл hello.html и добавьте в открывающую часть тега h1 атрибут style:

```
<h1 style="text-align: center">
```

Аналогично по центру расположите абзац простого текста в этом документе.

13. Вернитесь к файлу pagestyle.html. Содержимое тега style (без самого тега) перенесите в новый текстовый файл extstyle.css (в кодировке UTF-8).

В файле pagestyle.html исправьте секцию заголовка следующим образом:

```
4 <head>
5 <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
6 <title>Встроенные стили</title>
7 <link rel="stylesheet"
8     type="text/css"
9     href = "extstyle.css">
10 </head>
```

Исправленный HTML-документ сохраните под именем extstyle.html. Т.о. стили стали *внешними* по отношению к HTML-коду.

14. С помощью внешнего стилевого класса отобразите видимые границы ячеек в таблицах документа table.html (по умолчанию толщина границ равна 0 и границы невидимы).

### Псевдоклассы, псевдоэлементы и селекторы.

*Псевдоклассы* – механизм в CSS, позволяющий менять стиль объекта на странице в зависимости от положения объекта или его состояния (например, наведение курсора мыши на объект).

*Псевдоэлементы* – параметры стилей CSS, позволяющие применять их не ко всему объекту страницы, а к его части (например, только к первому символу в абзаце или только для нечетных строк в таблице).

*Селекторы* – способы *неявного* привязывания стилей CSS к объектам (например, стиль для объектов, следующих за определенным объектом, или сформированных только определенными тегами с определенным значением определенного атрибута).

15. В электронном справочнике прочитайте информацию о псевдоклассе :hover.

16. Реализуйте именованный внешний стиль, с помощью которого строки во второй таблице документа table.html при наведении курсора мыши окрашивались бы в желтый цвет.

### Г. Элементы формы.

Любой элемент, предполагающий взаимодействие с пользователем, должен быть размещен на *форме*, т.е. внутри парного тега <form>.

Для создания любого элемента на форме (текстового, элемента выбора – флажка или переключателя, кнопки и пр.) обычно используется универсальный объект <input>, который может принимать различный вид в зависимости от значения атрибута type.



## 17. Наберите HTML- код:

```

1  <!DOCTYPE html>
2  <!--Текстовые элементы-->
3  <html>
4  <head>
5      <title>Текстовые элементы форм</title>
6      <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
7  </head>
8  <body>
9      <h1>Текстовые элементы форм</h1>
10     <form>
11         Текстовое поле:
12         <input type = "text"
13             name = "txtInput"
14             value = "Текстовое поле">
15         <br>
16
17         Текстовая область:
18         <textarea name = "txtBigInput"
19             rows = 10
20             cols = 40>
21         Текстовая область
22     </textarea>
23     <br>
24
25     Пароль:
26     <input type = "password"
27         name = "secret"
28         value = "ха-ха!">
29     <br>
30
31     Скрытое поле: (Видишь его? Я тоже нет... А оно есть!)
32     <input type = "hidden"
33         name = "mystery"
34         value = "Скрытое поле">
35 </form>
36 </body>
37 </html>

```

Сохраните код в файле textForm.html и откройте в браузере. Прочитайте в электронном справочнике информацию о тегах <form> и <input>. Проследите, как в примере использован тег <input> с различными значениями атрибута type.

## 18. Создадим элементы выбора. Наберите следующий код:

```

1  <!DOCTYPE html>
2  <!--Элементы выбора-->
3  <html>
4  <head>
5      <title>Элементы выбора</title>
6      <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
7  </head>
8  <body>
9      <h1>Элементы выбора</h1>
10     <form>
11
12         флажки:
13         <input type = "checkbox"
14             name = "chkBurger">Да

```

```

15 <input type = "checkbox"
16     name = "chkFries">Нет
17 <input type = "checkbox"
18     name = "chkDrink">Кто знает?...
19 <br>
20 <br>
21
22 Переключатель
23 <input type = "radio"
24     name = "size"
25     value = "small">Маленький
26 <input type = "radio"
27     name = "size"
28     value = "medium">Средний
29 <input type = "radio"
30     name = "size"
31     value = "large">Большой
32 <br>
33 <br>
34
35 Раскрывающийся список
36 <select name = "selColor">
37     <option value = "red">red</option>
38     <option value = "orange">orange</option>
39     <option value = "yellow">yellow</option>
40     <option value = "green">green</option>
41     <option value = "blue">blue</option>
42     <option value = "indigo">indigo</option>
43     <option value = "violet">violet</option>
44 </select>
45 <br>
46 <br>
47
48 Множественный выбор
49 <select name = "lstColor"
50     size = 7
51     multiple>
52     <option value = "red">red</option>
53     <option value = "orange">orange</option>
54     <option value = "yellow">yellow</option>
55     <option value = "green">green</option>
56     <option value = "blue">blue</option>
57     <option value = "indigo">indigo</option>
58     <option value = "violet">violet</option>
59 </select>
60
61
62 </form>
63 </body>
64 </html>

```

Сохраните код и откройте его в браузере.

19. Создадим кнопки. Наберите следующий код:

```

buttonForm.html x
1 <!DOCTYPE html>
2 <!--Кнопки-->
3 <html>
4 <head>
5     <title>Кнопки</title>
6     <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
7 </head>

```

```

8  <body>
9  <h1>Кнопки</h1>
10 <form>
11 <textarea rows = 5>
12     Измените здесь текст, чтобы посмотреть, что происходит,
13     когда вы нажмёте на кнопку сброса.
14 </textarea>
15 <br><br>
16
17 <input type = "button"
18       value = "Просто кнопка">
19
20 <br><br>
21 <input type = "reset"
22       value = "Сброс">
23 <br><br>
24
25 <input type = "submit"
26       value = "Отправить">
27 <br><br>
28
29
30 </form>
31
32 </body>
33 </html>

```

Сохраните код в файле buttonForm.html, откройте его в браузере.

Почему не происходит отправка данных формы по нажатию кнопки «Отправить»?

20. Определите с помощью электронного справочника, что будет выведено на форму, если атрибут type тега <input> примет значение number? Какие браузеры поддерживают этот элемент формы?

Допишите соответствующий код в файл buttonForm.html и проверьте вывод этого элемента в вашем браузере.

### Контрольные вопросы:

- 1) Что такое World Wide Web?
- 2) Какие программные компоненты Web взаимодействуют посредством компьютерной сети Интернет? По каким протоколам?
- 3) Что необходимо браузеру, чтобы отобразить содержимое веб-страницы?
- 4) Откуда берёт код разметки HTTP-сервер?
- 5) Почему на веб-сервере не устанавливаются интерпретаторы кода HTML и CSS?
- 6) Перечислите основные положения концепции Web 2.0.
- 7) Что такое HTML?
- 8) Что такое тег HTML? Атрибут тега?
- 9) Что такое CSS? Почему стили называются каскадными?
- 10) Стил какого типа корректно использовать только один раз на веб-странице?
- 11) Какой псевдокласс позволит применять стиль только к нечетным строкам таблицы?

### Литература:

1. Мак-Дональд М. HTML5. Недостающее руководство, 2012.
2. Макфарланд Д. Большая книга CSS3. 3-е изд., 2014.
3. htmlbook.ru | Для тех, кто делает сайты [Электронный ресурс]. URL: <http://htmlbook.ru/> (дата обращения 25.03.2016).