

# ЛАБОРАТОРНАЯ РАБОТА 1

## Верстка макета веб-страницы с использованием языка HTML

**Целью** работы является получение практических навыков создания и форматирования веб-страниц с использованием средств разметки.

### Теоретическая часть

HTML — теговый язык разметки документов (*HyperText Markup Language*). Любой документ на языке HTML представляет собой набор элементов, причем начало и конец каждого элемента обозначается специальными пометками-тегами.

Основными компонентами HTML являются:

**Тег (tag).** Тег HTML это компонент, который командует Web-браузеру выполнить определенную задачу типа создания абзаца или вставки изображения.

**Атрибут (или аргумент).** Атрибут HTML изменяет тег. Например, можно выравнивать абзац или изображение внутри тега.

**Значение.** Значения присваиваются атрибутам и определяют вносимые изменения. Например, если для тега используется атрибут выравнивания, то можно указать значение этого атрибута. Значения могут быть текстовыми, типа *left* или *right*, а также числовыми, как например ширина и высота изображения, где значения определяют размер изображения в пикселях.

Теги могут быть:

1. обязательными (`<html>` `<head>` `<body>`)
2. парными (например `<h1></h1>`, `<div></div>`)
3. с атрибутами (`<a href="#" name="empty" target="_blank">Пустая ссылка </a>`)
4. пустыми (`<br>`, `<hr>`)

Основные правила синтаксиса HTML

– каждый тег должен начинаться с открывающейся угловой скобки, а заканчиваться закрывающейся угловой скобкой;

- браузеры игнорируют нестандартные теги и атрибуты, поэтому имена тегов и их атрибутов должны соответствовать перечню допустимых в HTML имен;
- при записи имен тегов можно пользоваться как верхним, так и нижним регистрами, но нельзя ставить пробелы;
- значения атрибутов можно записывать в любом регистре (за исключением пользовательских значений атрибутов, для которых соблюдается соответствие имен с точностью до регистра);
- если значение атрибута содержит пробелы, оно обязательно должно быть заключено в кавычки;
- в содержимом документа браузеры игнорируют несколько пробелов, следующих подряд, и сжимают их до одного пробела;
- элемент, включающий в себя начальный тег другого элемента, должен содержать и конечный тег этого элемента (за исключением одиночных тегов);
- допускается включать в HTML- документы комментарии.

#### *Ссылки*

Тег `<a>` является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия атрибутов `name` или `href` тег `<a>` устанавливает ссылку или якорь. Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы.

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. В качестве значения атрибута `href` используется адрес документа (URL, Universal Resource Locator, универсальный указатель ресурсов), на который происходит переход. Адрес ссылки может быть абсолютным и относительным. Абсолютные адреса работают везде и всюду независимо от имени сайта или веб-страницы, где прописана ссылка. Относительные ссылки, как следует из их названия, построены относительно текущего документа или корня сайта.

`<p>`

```
<a href="images/img.jpg">Ссылка на фото!</a>
```

</p>

<p>

<a href="page.html">Ссылка на страницу</a>

</p>

<a href="https://www.google.com.ua" name="link" target="\_blank">google</a>

### *Работа с изображениями html*



Тег <img> предназначен для отображения на веб-странице. Тег <img> предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Изображений в графическом формате GIF, JPEG или PNG. Адрес файла с картинкой задаётся через атрибут src. Если адрес файла с картинкой задаётся через атрибут src. Если необходимо, то рисунок можно сделать ссылкой на другой файл, поместив тег <img> в контейнер <a>. При этом вокруг файла, поместив тег <img> в контейнер <a>. При этом вокруг изображения отображается рамка, которую можно убрать, добавив атрибут border="0" в тег <img>. 

### *Работа со списками html*

<ul>

<li>Заголовок должен быть короче трех строк.</li>

<li>При названии разделов используйте уже устоявшиеся термины, такие как гостевая книга, чат, ссылка, главная страница и другие.

</li>

<li>Перед использованием специального термина или слова, решите, будет ли оно понятно читателю.

</li>

</ul>

### *Тег <meta>*

<meta> определяет метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатегам для

получения описания сайта, ключевых слов и других данных. Разрешается использовать более чем один метатег, все они размещаются в контейнере `<head>`. Как правило, атрибуты любого метатега сводятся к парам «имя=значение», которые определяются ключевыми словами `content`, `name` или `http-equiv`.

Кодировка:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" >
```

Описание:

```
<meta name="description" content="Сайт об HTML и создании сайтов" >
```

Ключевые слова:

```
<meta name="Keywords" content="HTML, META, метатег, тег, поисковая система">
```

*Таблицы (Тег `<table>`)*

Элемент `<table>` служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов `<tr>` и `<td>`. Внутри `<table>` допустимо использовать следующие элементы: `<caption>`, `<col>`, `<colgroup>`, `<tbody>`, `<td>`, `<tfoot>`, `<th>`, `<thead>` и `<tr>`.

Таблицы с невидимой границей долгое время использовались для верстки веб-страниц, позволяя разделять документ на модульные блоки. Подобный способ применения таблиц нашел воплощение на многих сайтах, пока ему на смену не пришел более современный способ верстки с помощью слоев.

*div & span*

Элемент `r` имеет семантическое значение. Он определяет абзац текста.

Такой элемент не стоит использовать там, где это по смыслу не имеет значения. Например, не стоит помещать в `r` меню с кнопками для навигации по сайту.

Два самых используемых элемента — `div` и `span` — не имеют семантического значения. Они нужны исключительно для структуры и стилей.

`div` — это block-level элемент, у него свойство `display: block`.

`span` — это inline-level элемент, у него свойство `display: inline`.

Конечно, вы можете изменить эти свойства в своем CSS, но лучше так не делать. В целом, лучше не менять базовые свойства у типов элементов, а добавлять классы по необходимости.

div обычно используется для определения какого-то крупного блока на странице. А span — для небольшого элемента на строке.

```
<div class="info">
  <p>
    После Второй мировой войны <span class="city">Нью-
    Йорк</span> стал одним из неоспоримых мировых
    городов-лидеров.
  </p>
  <p>
    Строительство штаб-квартиры ООН в Нью-Йорке
    символизировало уникальное политическое значение
    города.
  </p>
</div>
```

## Ход работы

1. Создать директорию для проекта с именем «Фам\_Гр».
2. В директории создать файл с именем «index.html». Открыть его помощью редактора Notepad++.

3. Ввести базовую структуру страницы:

```
<!doctype>
<html>
  <head>
    <title>Лабораторная работа 1</title>
  </head>
  <body>
    <h1>Заголовок</h1>
    <p>Текст</p>
  </body>
</html>
```

Привести введенный код верстки в отчете к лабораторной работе.

4. Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

5. Для заданного макета верстки составить перечень составных блоков. Например, «верхний колонтитул», «логотип», «меню», «пункт меню 1» и пр. В отчете создать таблицу следующего вида:

№	Вид блока	ID блока	HTML-код
1	Верхний колонтитул	Header	<div id="header">...</div>
2	...	...	...

6. В соответствии с макетом страницы последовательно создать элементы блочной верстки div. Например,

```
<div id="main">Главное меню:  
  <div id="content">  
    Некоторый текст для проверки размещения элемента.  
  </div>  
</div>
```

Привести введенный код верстки в соответствующих столбцах таблицы из п. 5 в отчете к лабораторной работе.

7. Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

8. Проверить валидность созданного HTML-файла с помощью сервиса <http://validator.w3.org/>. Включить в отчет сообщения об ошибках. Исправить выявленные ошибки, провести повторную валидацию. Полученные результаты также добавить в отчет.

### **Задание на выполнение**

В соответствии с выданным преподавателем вариантом подготовить HTML-верстку веб-страницы. По мере выполнения работы сформировать отчет, содержащий описание структуры страницы, реализующий ее программный код верстки и экранные формы полученного результата.

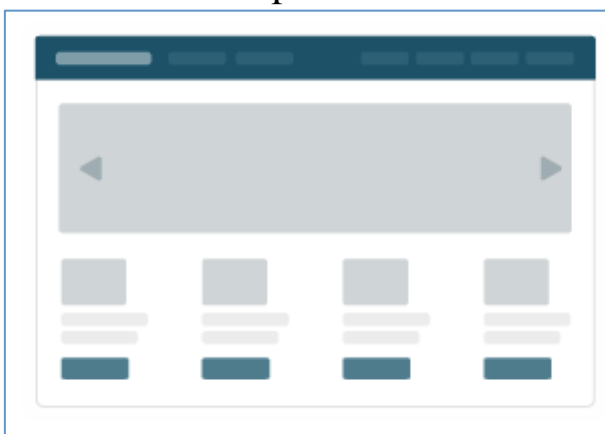
## Варианты заданий



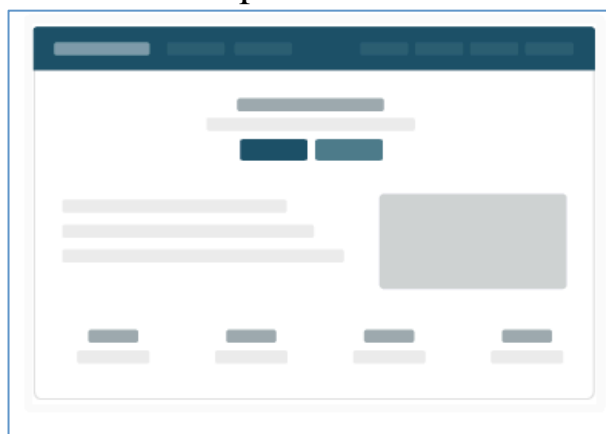
Вариант 1



Вариант 2



Вариант 3



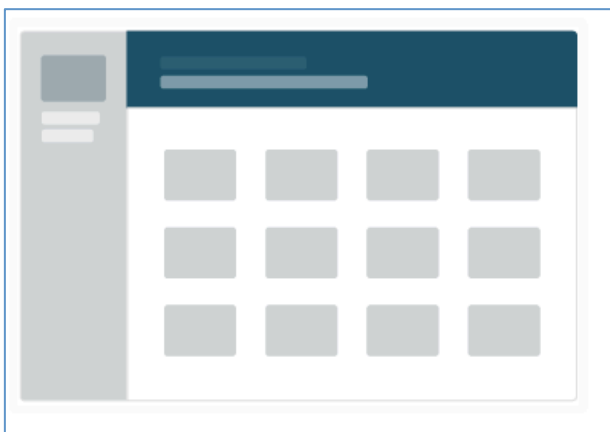
Вариант 4



Вариант 5



Вариант 6



Вариант 7



Вариант 8

### **Требования к отчету**

Отчет должен содержать все элементы, описанные в разделе «Ход выполнения работы».

### **Контрольные вопросы**

1. Что такое HTML?
2. Какова базовая структура верстки?
3. Для чего используется элемент div?
4. Для чего используется элемент span?
5. Для чего используется элемент p?
6. Для чего используется элемент head?



## ЛАБОРАТОРНАЯ РАБОТА 2

### Стилизация макета веб-страницы с использованием языка CSS

**Целью** работы является получение практических навыков форматирования веб-страниц с использованием таблиц стилей CSS.

#### Теоретическая часть

CSS (Cascading Style Sheets, каскадные таблицы стилей) – представляет собой набор стилевых описаний имеющихся в документе html-тегов. Каждое подобное описание может быть сопоставлено одному из элементов документа, нескольким элементам, удовлетворяющим некоторому общему условию или даже всем элементам документа. CSS является дополнением языка HTML и развивается параллельно с ним, постоянно увеличивая количество свойств и мощь функциональных возможностей. CSS позволяет задать для элементов документа такие параметры отображения, как размер и начертание шрифта, поля, размеры и положение на странице, фоны и картинки, анимировать изменение параметров отображения, адаптировать внешний вид документа к условиям отображения, то есть решает вопросы формирования представления элементов в окне браузера.

Основные преимущества CSS:

- отделение представления от разметки позволяет гибче настраивать внешний вид документа;
- изменение стилевого оформления не влияет на разметку;
- стилевое оформление позволяет адаптировать внешний вид к носителю, на котором будет отображаться страница;
- стилевое оформление позволяет вносить динамику в отображение документа (например, псевдокласс :hover или свойство CSS3 animation);
- иерархия описания стилей позволяет наследовать способы представления элементов от контейнера к внутренним элементам;
- возможность определения альтернативных стилей для документа позволяет варьировать его внешний вид.

Таблицы стилей строятся как совокупность правил. Каждое правило описывает правила отображения того или иного документа. К какому(-им) элементу(-ам) относится то или иное правило, определяют один или несколько селекторов. Блок определения правила выделяется фигурными скобками и содержит набор свойств, отделенных друг от друга точкой с запятой (;) (после последнего свойства точка с запятой необязательна). Для каждого свойства указывается значение, отделенное от названия свойства знаком двоеточия (:).

Виды селекторов:

1. Селекторы тегов. Если в качестве селектора правила указать наименование тега, то оно будет применяться ко всем элементам, добавленным в документ с использованием данного тега.

2. Селектор-идентификатор. Селектор данного типа начинается с символа '#', и связанное с ним правило будет применено к элементу, имеющему аналогичное значение в атрибуте стилевого идентификатора id. Если учесть, что два разных элемента на странице не могут иметь одинаковый атрибут id, можно утверждать, что подобным образом оформляются уникальные стили элементов документа.

3. Селектор класса. Данный вид селекторов позволяет сопоставить правилу оформления группу элементов, причем необязательно с одним и тем же тегом. Главное, чтобы у всех этих элементов был общий атрибут стилевой класс class.

В таблице стилей правило для класса задаётся с использованием символа «.».

4. Селекторы псевдоклассов. Псевдоклассы привязывают правило к некоторому состоянию элемента. Если элемент находится в этом состоянии, то правило к нему применяется. Примерами состояний могут быть: нахождение курсора мыши над элементом, получение элементом фокуса ввода, нахождение элемента в определенной позиции иерархии DOM элемента-родителя.

В связи с тем, что состояние изменчиво, правило может применяться к элементу не постоянно: пользователь увел курсор с элемента – состояние изменилось, правило больше неактуально.

Объявление селектора с псевдоклассом включает название некоторого селектора и имя псевдокласса через символ «:».

5. Универсальный селектор «\*» означает, что данное правило будет применено ко всем элементам.

В названии технологии присутствует слово «каскадных». В чем заключается идея каскадности стилей? Каскадность означает, что значения для свойств элемента может быть сформирована не одним правилом, а целой совокупностью правил. Подобная ситуация может возникнуть в случаях:

- использования различных таблиц стилей, которые для одних и тех же селекторов задают свои правила отображения;
- наложения для одного элемента нескольких правил по разным селекторам (по тегу, стилевому идентификатору, стилевому классу);
- наследования значений свойств дочерних элементов от родительских в дереве иерархии DOM.

Если к конкретному элементу веб-страницы применено несколько стилей, то свойства объединятся при условии, что они не конфликтуют между собой

## **Ход работы**

1. В директорию проекта, созданную в рамках предыдущей лабораторной работы, записать файл с именем «style.css».

2. Подключить таблицу стилей к имеющейся HTML-странице. Для этого с помощью текстового редактора Notepad++ между открывающим и закрывающим тегами элемента «head» ввести следующую строку:

```
<link rel="stylesheet" href="style.css"/>
```

3. В HTML-заготовку в код разметки заголовка страницы добавить двухуровневое меню (число пунктов зависит от задания):

```
<div id="menuwrapper">
  <ul>
    <li><a href="#">Button 1</a></li>
    <li><a href="#">Button 2</a>
      <ul>
        <li><a href="#">Button 2.1</a></li>
```

```

        <li><a href="#">Button 2.2</a></li>
    </ul>
</li>
</ul>
</div>

```

4. В таблицу стилей «style.css» добавить описание стиля оформления пунктов меню первого уровня (размер и начертание шрифтов, величину и расцветку блоков можно менять по своему усмотрению):

```

#menuwrapper ul li:hover{
    background-color:#6679e9;
    position:relative;
}
#menuwrapper ul li a{
    padding:0px 15px;
    color:#ffffff;
    display:inline-block;
    text-decoration:none;
}

```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

5. В таблицу стилей «style.css» добавить описание стиля оформления пунктов меню второго уровня (размер и начертание шрифтов, величину и расцветку блоков можно менять по своему усмотрению):

```

#menuwrapper ul li ul{
    position:absolute;
    display:none;
}
#menuwrapper ul li:hover ul{
    left:0;
    top:25px;
    display:block;
}

```

```
#menuwrapper ul li ul li {
    float:none;
    width:150px;
    background-color:#cae25a;
}
#menuwrapper ul li:hover ul li:hover{
    background-color:#b1b536;
}
#menuwrapper ul li ul li a{
    color:#454444;
    display:inline-block;
    width:120px;
}
```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

6. Аналогичным образом добавить hover-эффект для всех меню шаблона (при наличии). Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

7. Для любого квадратного (или прямоугольного) блока (или нескольких) на странице добавить эффект поворота на 360° при наведении на него курсора.

Пусть в HTML-заготовке блока закодирован как:

```
<div id="test1">Box</div>
```

Тогда в таблицу стилей следует добавить следующий код:

```
#test1 {
    transition: 0.70s;
    -webkit-transition: 0.70s;
    -moz-transition: 0.70s;
    -ms-transition: 0.70s;
    -o-transition: 0.70s;
    width: 250px;
    height: 150px;
    display: block;
```

```

margin-right: auto;
margin-left: auto;
background-color:red; width:100px; color:white; height:100px
}
#test1:hover {
transition: 0.70s;
-webkit-transition: 0.70s;
-moz-transition: 0.70s;
-ms-transition: 0.70s;
-o-transition: 0.70s;
-webkit-transform: rotate(360deg);
-moz-transform: rotate(360deg);
-o-transform: rotate(360deg);
-ms-transform: rotate(360deg);
transform: rotate(360deg);
}

```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

8. Добавить стилизованную строку поиска в заготовку HTML. Для этого

8.1 В HTML-файл подключить таблицу стилей для Font Awesome через CDN:

```

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css"/>

```

8.2 Добавить код разметки поля поиска:

```

<div class="search">
  <i class="fa fa-search left-icon"></i>
  <input placeholder="Search here" />
</div>

```

8.3 В таблицу стилей добавить код:

```

.search input {
width:55px; height:20px;
border:1px solid #ccc;
padding-left:21px;
}

```

```
padding-right:30px;
color:black;
font-family: arial,sans-serif;
font-size: 10px;
}
.search .left-icon {
position:absolute;
left:15px;
top:15px;
}
.search .fa {
color:#444;
font-size: 12px;
}
```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

### **Задание на выполнение**

В соответствии с выданным преподавателем вариантом подготовить CSS-таблицы стилей веб-страницы. По мере выполнения работы сформировать отчет, содержащий описание структуры страницы, реализующий ее программный код верстки и экранные формы полученного результата.

### **Требования к отчету**

Отчет должен содержать все элементы, описанные в разделе «Ход выполнения работы».

### **Контрольные вопросы**

7. Что такое CSS?
8. Какова базовая структура CSS-описания?

9. Какие селекторы могут быть использованы в CSS?
10. Что означает «каскадность» таблиц стилей?
11. Как подключить таблицу стилей к HTML-странице?
12. Как решаются конфликты стилей?

## **ЛАБОРАТОРНАЯ РАБОТА 3**

### **Интерактивность веб-страницы с использованием библиотеки jQuery UI**

**Целью** работы является получение практических навыков обеспечения интерактивности веб-страниц с использованием библиотеки jQuery.

#### **Теоретическая часть**

jQuery - это библиотека, которая значительно упрощает и ускоряет написание JavaScript кода.

Девиз jQuery "write less, do more" (пиши меньше, делай больше) отражает ее главное предназначение.

jQuery позволяет создавать анимацию, обработчики событий, значительно облегчает выбор элементов в DOM и создание AJAX запросов.

Данная библиотека работает со всеми браузерами. Это значит, что Вам больше не нужно будет беспокоиться о кроссбраузерной совместимости JavaScript кода.

Для jQuery написано огромное количество плагинов, которые позволяют расширить ее возможности еще больше.

jQuery доступна в сжатом и несжатом варианте. Несжатую версию удобно использовать во время разработки и отладки вашего сайта. Сжатая версия практически бесполезна при отладке, однако она значительно меньше весит и поэтому ее целесообразно применять на выпущенной версии сайта, для более оптимальной работы (сжатый скрипт будет быстрее подключен к странице, а так же быстрее обработан).



Существует альтернативный вариант подключить jQuery на страницы вашего сайта (не закачивая библиотеку на сервер). Можно подключить библиотеку, которая находится не на вашем сервере, а на серверах CDN. Существуют несколько таких хранилищ, наиболее известные и надежные из них Google CDN, Microsoft CDN, а так же CDN который организовали создатели jQuery.

Для того чтобы понимать как работает селектор необходимы базовые знания CSS, т.к. именно от принципов CSS отталкивает селектор jQuery:

- `$("#header")` — получение элемента с `id=«header»`
- `$(<h3>)` — получить все `<h3>` элементы
- `$(<div#content .photo>)` — получить все элементы с классом `=«photo»` которые находятся в элементе `div` с `id=«content»`
- `$(<ul li>)` — получить все `<li>` элементы из списка `<ul>`
- `$(<ul li:first>)` — получить только первый элемент `<li>` из списка `<ul>`

jQuery UI — это библиотека на основе jQuery, реализующая более 20 плагинов, среди которых плагины организующие различное поведение (например перетаскивание или растягивание элементов), восемь видов виджетов (такие как календарь, диалоговые окна, систему вкладок и т.д) и анимационные эффекты. Кроме этого, UI обладает несколькими темами оформления, с помощью которых оформляются виджеты и которые содержат набор полезных иконок (173 штуки). Любая из тем оформления может быть подкорректирована прямо на сайте jQuery UI, непосредственно перед загрузкой.

## Ход работы

1. Подключить к имеющейся HTML-странице библиотеки и плагины JQuery. Для этого с помощью текстового редактора Notepad++ между открывающим и закрывающим тегами элемента `<head>` ввести следующие строки:

```
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>  
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
```

Здесь же добавить ссылку на используемые стили:

```
<link  
  rel="stylesheet"  
  href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css"/>
```

2. В HTML-заготовку в код разметки поля ввода (для поиска из предыдущей работы) добавить всплывающую подсказку с помощью атрибута title (значение атрибута указать самостоятельно).

Также между открывающим и закрывающим тегами элемента «head» ввести следующие строки:

```
<script>  
  $( function() {  
    $( document ).tooltip();  
  } );  
</script>
```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

3. Добавить на страницу числовое поле ввода с возможностью инкрементирования / декрементирования значения как в самом элементе на форме, так и с клавиатуры. Для этого в HTML-заготовку в код разметки любого блока «div» добавить разметку поля ввода с идентификатором «spin».

Далее после строки «\$( document ).tooltip();» добавить код:

```
var spinner = $( "#spin" ).spinner();
```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

4. Добавить на страницу текстовое поле ввода (идентификатор «tags») с автозаполнением. Для этого после строки `var spinner = $( "#spin" ).spinner();` добавить код:

```
var availableTags = [  
    //в двойных кавычках через запятую ввести 20 ключевых слов  
    одной тематики  
];  
$( "#tags" ).autocomplete({  
    source: availableTags  
});  
});
```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

5. Добавить на страницу текстовое поле выбора даты. Для этого в HTML-заготовку в код разметки любого блока «div» добавить разметку поля ввода с идентификатором «datepicker».

Далее после строки `$( document ).tooltip();` добавить код:  
`$( "#datepicker" ).datepicker();`

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

6. Добавить на страницу элемент управления «аккордион». Для этого в HTML-заготовку в код разметки любого блока «div» добавить идентификатор «accordion». В структуру блока добавить разделы аккордиона:

```
<h3>Section 1</h3>  
<div>  
    <p>
```

Sed non urna. Donec et ante. Phasellus eu ligula. Vestibulum sit ame purus. Vivamus hendrerit, dolor at aliquet laoreet, mauris turpis porttitor velit, faucibus interdum tellus libero ac justo. Vivamus non quam. In suscipit faucibus urna. </p>

</div>

<h3>Section 2</h3>

<div>

<p>

Sed non urna. Donec et ante. Phasellus eu ligula. Vestibulum sit ame purus. Vivamus hendrerit, dolor at aliquet laoreet, mauris turpis porttitor velit, faucibus interdum tellus libero ac justo. Vivamus non quam. In suscipit faucibus urna.

</p>

</div>

и т.д.

Далее после строки «\$( document ).tooltip();» добавить код:  
\$( "#accordion" ).accordion();

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

## **Требования к отчету**

Отчет должен содержать все элементы, описанные в разделе «Ход выполнения работы».

## **Контрольные вопросы**

13. Что такое JQuery?
14. Как подключить JQuery к проекту?
15. Какие селекторы могут быть использованы в JQuery?
16. Что такое CDN?

17. Приведите примеры плагинов ?
18. Как решаются конфликты скриптов?

## **ЛАБОРАТОРНАЯ РАБОТА 4**

### **Использование API картографических сервисов при разработке веб-приложений**

**Целью** работы является получение практических навыков применения картографических данных в веб-приложениях с использованием средств API.

#### **Теоретическая часть**

Картографический сервис — это специализированная информационная система, предоставляющая пространственные данные в виде интерактивной карты. Картографический веб-сервис обеспечивает веб-доступ к картографической информации на основе интерфейсов прикладного программирования (API). В настоящее время на российском рынке наиболее известны и распространены следующие картографические и справочные сервисы:

- Яндекс.Карты;
- Google Maps;
- ГИС;
- ArcGIS.

Архитектуру ArcGIS для разработчиков можно представить как серию клиентов различных типов (рабочих столов, мобильных устройств, веб-браузеров и т. д.) и серверных подключений через простую сервисную модель.

ArcGIS включает в себя несколько приложений клиентов для настольных компьютеров, интернета и мобильных сред, обеспечивающих доступ и применение данных и сервисов ГИС, расположенных в облаке или на корпоративных серверах. Кроме того, для каждого типа клиента доступны API разработчика, позволяющие внедрять собственные приложения и рабочие алгоритмы пользователя.

При помощи ArcGIS разработчики могут получать доступ и работать с рядом Web API для JavaScript, Adobe Flex и Microsoft Silverlight, чтобы строить пользовательские веб-приложения для использования с сервисами ArcGIS. Инфраструктура этих веб-приложений подключается и использует картографические и прочие службы ГИС, опубликованные при помощи ArcGIS for Server, ArcGIS Online и открытых пользовательских картографических серверов в Интернете. Эти API используют открытую архитектуру на основе REST, подчиняющуюся стандартам индустрии, которая позволяет содержанию ArcGIS и сервисам быть использованным, как составная часть ArcGIS, а также вне ArcGIS.

## Ход работы

1. Подключить к имеющейся HTML-странице стили и скрипты ArcGIS API for JavaScript. Для этого с помощью текстового редактора Notepad++ между открывающим и закрывающим тегами элемента «head» ввести следующие строки:

```
<link rel="stylesheet"
      href="https://js.arcgis.com/4.9/esri/css/main.css"/>
<script src="https://js.arcgis.com/4.9/"></script>
```

2. В HTML-заготовке из предыдущей работы выбрать любой «свободный» элемент div и присвоить его атрибуту id значение map.

3. Загрузить из API модули для создания карты и ее визуализации в двумерном формате. Для этого между открывающим и закрывающим тегами элемента «head» ввести следующие строки:

```
<script>
  require([
    "esri/Map",
    "esri/views/MapView"
  ], function(Map, MapView) {
```

```
//  
});  
</script>
```

4. Создать экземпляр класса Map (карта). Для этого заменить символ «`//`» из предыдущего пункта следующим фрагментом:

```
var map = new Map({  basemap: "streets"  });
```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

5. Привязать экземпляр карты к элементу div с id = "map". Для этого к предыдущему фрагменту добавить следующий код:

```
var view = new MapView({  
  container: "map",  
  map: map  
});
```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

Изучить другие варианты базовых карт: satellite, hybrid, topo, gray, dark-gray, oceans, osm, national-geographic (изменять значение атрибута basemap). Сделать снимки экрана браузера и добавить их в отчет к лабораторной работе.

6. Добавить к созданной карте параметр масштаба, заданный по умолчанию. Для этого к конструктору MapView добавить параметр:

```
zoom: 4,
```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

Протестировать параметр масштаба в диапазоне от 1 до 12. Сделать снимки экрана браузера и добавить их в отчет к лабораторной работе.

7. Центрировать карту в соответствии с заданными значениями географических широты и долготы. Для этого к конструктору MapView добавить параметр:

center: [15, 65]

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

Определить параметры своего местоположения любым известным способом. Ввести полученные значения в качестве параметров свойства center. Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

### **Требования к отчету**

Отчет должен содержать все элементы, описанные в разделе «Ход выполнения работы».

### **Контрольные вопросы**

19. Что такое картографический сервис?
20. Приведите примеры использования картографических сервисов?
21. Что такое API?
22. Как подключить ArcGIS API for JavaScript к веб-приложению?
23. К какому элементу страницы применяются свойства и методы ArcGIS API for JavaScript?



## **ЛАБОРАТОРНАЯ РАБОТА 6**

### **Использование фреймворка Bootstrap при разработке веб-приложений**

**Целью** работы является получение практических навыков применения фреймворка Bootstrap при разработке веб-приложений.

#### **Теоретическая часть**

Bootstrap (также известен как Twitter Bootstrap) — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

Основные инструменты Bootstrap:

- Сетки — заранее заданные размеры колонок, которые можно сразу же использовать, например ширина колонки 140 px относится к классу `.span2` (`.col-md-2` в третьей версии фреймворка), который можно использовать в CSS-описании документа.
- Шаблоны — фиксированный или резиновый шаблон документа.
- Типографика — описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т. п.
- Медиа — представляет некоторое управление изображениями и видео.
- Таблицы — средства оформления таблиц, вплоть до добавления функциональности сортировки.
- Формы — классы для оформления форм и некоторых событий, происходящих с ними.
- Навигация — классы оформления для панелей, вкладок, перехода по страницам, меню и панели инструментов.
- Алерты — оформление диалоговых окон, подсказок и всплывающих окон.

Система сеток в Bootstrap используется для шаблона, а в частности для создания адаптивных шаблонов. Сетка сделана из группирования

строк и колонок внутри одного или нескольких контейнеров. В сетке фреймворка по умолчанию 12 колонок.

Вся сетка должна располагаться в общем контейнере. Это должен быть блок с классом `container` или `container-fluid`. Отличие у классов в том, что первый имеет максимальные фиксированные размеры, а именно 1170 пикселей. То есть ширина сайта не будет больше этого значения.

`Container-fluid` – это полностью резиновый контейнер, который растягивается всегда на 100% ширины окна, поэтому если у вас полностью резиновый сайт, вам нужен именно такой контейнер. Если мы говорим о таблицах, то там тоже есть свой глобальный контейнер – `table`.

Внутри этого контейнера должен быть еще один блок с классом `row`, то есть одна строка сетки. И опять же, если сравниваем с таблицами, то там за вывод одного табличного ряда отвечает тег `tr`. Ряды сетки и таблицы очень похожи, потому что в обоих случаях они просто выступают контейнерами для основного содержимого – в них самих никакой контент не размещается, да и стили к ним не применяются.

В самом ряду располагаются уже непосредственно ячейки или колонки, как принято говорить при верстке на Bootstrap. Колонка имеет класс `col-x-x`, где первый `x` – обозначение устройства, а второй – кол-во колонок от 1 до 12.

## Возможности сетки

Полезная таблица, показывающая, как аспекты сетки Bootstrap работают на различных устройствах.

	Очень маленькие устройства Телефоны (<768px)	Маленькие устройства Планшеты (≥768px)	Средние устройства Компьютеры (≥992px)	Большие устройства Компьютеры (≥1200px)
Работа сетки	Всегда горизонтальная	Сначала сжатая, горизонтальная над контрольной точкой		
Ширина контейнера	Нет (авто)	750px	970px	1170px
Префикс класса	<code>.col-xs-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>
# of columns	12			
Ширина колонок	Авто	~62px	~81px	~97px
Ширина интервала	30px (15px с каждой стороны колонки)			
Вложения	Да			
Отступ	Да			
Порядок колонок	Да			

## Пример хода работы

1. На официальной странице Bootstrap (<https://getbootstrap.com/docs/4.1/getting-started/download/>) загрузить полную версию таблиц стилей и сценариев (Compiled CSS and JS).

2. Распаковать полученный архив. Директорию переименовать в lab5.

3. Создать файл index.html (в кодировке utf-8).

4. Открыть созданный файл и прописать следующий код:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap 101 Template</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet" media="screen">

    <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements
and media queries -->
    <!--[if lt IE 9]>
      <script src="../../assets/js/html5shiv.js"></script>
      <script src="../../assets/js/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="//code.jquery.com/jquery.js"></script>
    <!-- Include all compiled plugins (below), or include individual files
as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
```

</html>

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

5. Создать на странице строку-заголовок с контейнером, содержащим логотип (произвольный) и 4 пункта меню («Home», «About», «Services», «Contact»).

5.1 Создать строку-панель навигации. Для этого заменить строку «<h1>Hello, world!</h1>» на фрагмент:

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
</nav>
```

5.2 Добавить элемент-контейнер. Для этого между открывающим и закрывающим тегами элемента nav ввести код:

```
<div class="container">
</div>
```

5.3 Добавить логотип в строку (генерация логотипа-плейсхолдера через сервис <http://placeholder.it>). Для этого между открывающим и закрывающим тегами контейнера из предыдущего пункта добавить:

```
<a href="#">
  
</a>
```

5.4 Разместить пункты меню в строке. Для этого после разметки логотипа из предыдущего пункта (но внутри элемента-контейнера) разместить код:

```
<div class="collapse navbar-collapse" id="navbarResponsive">
  <ul class="navbar-nav ml-auto">
    <li class="nav-item active">
      <a class="nav-link" href="#">Home
        <span class="sr-only">(current)</span>
      </a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">About</a>
```

```

</li>
<li class="nav-item">
  <a class="nav-link" href="#">Services</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="#">Contact</a>
</li>
</ul>
</div>

```

5.4 Ввести разметку «свертки» меню. Для этого после разметки логотипа (но внутри элемента-контейнера) ввести код:

```

<button      class="navbar-toggler"      type="button"      data-
toggle="collapse"      data-target="#navbarResponsive"      aria-
controls="navbarResponsive"      aria-expanded="false"      aria-label="Toggle
navigation">
  <span class="navbar-toggler-icon"></span>
</button>

```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

6. Добавить на страницу три равноразмерных блока по сетке Bootstrap.

6.1 Для этого после строки навигации из предыдущего пункта добавить элемент-контейнер (по аналогии с предыдущим пунктом).

6.2 Создать строку для разметки блоков (внутри контейнера):

```

<div class="row">
</div>

```

6.3 Ввести разметку трех блоков (внутри строки):

```

<div class="col-sm">
  One of three columns
</div>
<div class="col-sm">

```

```
    One of three columns
</div>
<div class="col-sm">
    One of three columns
</div>
```

Открыть страницу в браузере, проверить работоспособность кода. Сделать снимок экрана браузера и добавить его в отчет к лабораторной работе.

### **Задание на работу**

1. С помощью Bootstrap самостоятельно воспроизвести макет страницы из первой лабораторной работы.
2. Самостоятельно изучить реализацию в Bootstrap и добавить на страницу:
  - календарь
  - слайдер
  - строку поиска
  - автозаполнение

### **Требования к отчету**

Отчет должен содержать все элементы, описанные в разделе «Ход выполнения работы».

### **Контрольные вопросы**

24. Что такое API?
25. Как подключить Bootstrap к проекту?
26. Что представляет собой сетка Bootstrap?