

Лабораторная работа № 31

Тема работы: «Подключение библиотеки jQuery к приложению и получение доступа к элементам документа»

1. Цель работы

Формирование умений подключения библиотеки JQuery и использования различных методов получения доступа к html-элементам.

2. Задание

Выполнить задания в соответствии с порядком выполнения работы средствами библиотеки JQuery.

3. Оснащение работы

ПК, редактор исходного кода, браузер.

4. Основные теоретические сведения

jQuery — библиотека, которая позволяет делать код короче, а также позволяет внутри страницы настроить код, который бы срабатывал как триггер (предопределенный набор действий, который выполняется автоматически при наступлении связанного с ним события, если этот код описываем в области <head> ... </head>).

jQuery библиотека содержит следующий функционал:

- операции с HTML/DOM (манипулирование компонентами HTML/DOM)
- операции с CSS-селекторами
- HTML-обработчики событий
- Эффекты анимации
- AJAX
- Utilities.

jQuery упрощает работу с JavaScript, а также вызовы AJAX и DOM-манипуляции. Есть много фреймворков JavaScript, но JQuery, является самым популярным и используемым за счет своей расширяемости.

Библиотеку jQuery можно скачать с сайта <http://jquery.com>, а можно вставить в документ, используя известные интернет-адреса:

- по адресу <http://code.jquery.com/jquery-latest.js> — доступна всегда последняя версия.
- с Google jquery можно загрузить любую из не слишком старых версий: <https://developers.google.com/speed/libraries/devguide?hl=ru#>. Синтаксис такой: `src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"`, где 1.8.3

— версия, причём можно указать её приблизительно: 1.8 означает последнюю версию вида 1.8.*, а 1 — последнюю версию вида 1.*. Файл jquery.min.js обозначает сжатый код, а jquery.js — несжатый, для удобства отладки;

— либо с Microsoft CDN: src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.9.1.min.js"

Пример подключения ниже (использовать один из вариантов подключения библиотеки), рисунок 31.1.

```
<!DOCTYPE html>
<html>
<head>
<!--если выбираем библиотеку от Microsoft CDN -->
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.9.1.min.js"></script>
<!--если выбираем библиотеку от Google -->
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<!--третий вариант -->
<script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
<body>
</body>
</html>
```

Рисунок 31.1 – Способы подключения библиотеки jQuery

Базовая команда для библиотеки выглядит как: \$(селектор).action(), где

— \$ - предписание использовать jQuery;

— (селектор) - это "запрос или элементы поиска" в HTML элементах страницы;

— action() - это действия, которые должны быть выполнены над найденными элементами (это те элементы, которые удовлетворяют условиям селектора).

Например:

\$(this).hide() – скрывает текущий элемент (где this – это указатель на текущий элемент, позволяет делать код универсальным за счет того, что не надо писать здесь имя или id элемента, над которым будет производиться действие hide()).

\$("p").hide() – скрывает все <p> элементы на странице.

\$(".test").hide() – скрывает все элементы на странице, которые ассоциированы с классом "test".

\$("#test").hide() – скрывает все элементы на странице, у которых id="test".

В большинстве примеров jQuery-методы находятся внутри события документа Ready():

```
$(document).ready(function(){
    // jQuery-методы размещаем здесь...
});
```

Это необходимо для предотвращения любых срабатываний JQuery-кода, прежде чем документ не закончит полную загрузку. Это хорошая практика, чтобы дождаться, пока документ будет полностью загружен и готов до работы с ним. Это также позволяет вам сформировать свой JavaScript код в головной части, прежде чем тело документа.

JQuery селекторы позволяют делать выборку (поиск) и манипулировать с элементами HTML. Селекторы по сути это набор условных обозначений и правил для выборки и манипулирования (в конце лабораторной в приложении дан большой список примеров селекторов).

С JQuery селекторами вы можете найти элементы страницы, основанные на идентификаторе id, классах (class), типах (type), атрибутах (attribute), значениях атрибутов (value) и др. Также они базируются и на CSS Selectors, в дополнении вы можете создать свой селектор. Все типы селекторов в JQuery начинаются с указания \$ и парных скобок: \$()

Например, в следующем коде при нажатии на кнопку выполняется поиск на странице всех элементов, обозначенных тегом <p> , и все эти элементы скрываются на странице (срабатывает метод hide()), рисунок 31.2

```
<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js">
</script>

<script>
$(document).ready(function () {
    $("button").click(function () {
        $("p").hide();
    });
});
</script>
</head>

<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

Рисунок 31.2 – Работа с методом hide()

Пояснения к скрипту:

<script>

//в строке ниже в качестве селектора использован весь объект –документ,
//при этом будет срабатывать jquery-запрос, как только наступит событие
полной готовности

//страницы к работе с пользователем и запустится метод ready(),

// при срабатывании которого будет создана следующая функция

\$(document).ready(function () {

//созданная функция в свою очередь будет jquery-запросом, который ищет все элементы типа

//button – кнопка, и с их методами click связывает (ассоциирует) действие в виде функции,

```
$("#button").click(function () {
```

// которая выполнит jquery-запрос, который для всех найденных элементов внутри тега <p> выполнит метод hide(), т.е. скроет их со страницы

```
$("#p").hide();
```

```
//далее закрываем внутреннюю функцию
```

```
});
```

```
//далее закрываем внешнюю функцию
```

```
});
```

```
</script>
```

В таком исполнении скрипт, помещенный в заголовок страницы внутрь метода объект document.ready() работает как триггер, т.е. автоматически срабатывает при наступлении определенного события на странице. В примере – это событие click() кнопки. А так как в методе hide() нет никакого описания другого кода, то он выполняет те действия, для которых он изначально создан, а именно скрывает объект.

Селектор jQuery #id использует id атрибут в HTML-тегах, чтобы найти определенный элемент. Id должен быть уникальным внутри всей страницы, если вы хотите найти с его помощью конкретный уникальный элемент. Чтобы найти элемент с помощью id, то перед названием искомого идентификатора ставится знак #, например: \$("#test")

При нажатии на кнопку ищется элемент с идентификатором test и скрывается со страницы, рисунок 31.3.

```
<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js">
</script>

<script>
$(document).ready(function () {
    $("#button").click(function () {
        $("#test").hide();
    });
});
</script>
</head>

<body>
<h2>This is a heading 2</h2>
<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>
<button>Click me</button>
</body>

</html>
```

Рисунок 31.3 – Работа метода hide()

На рисунке 31.4 приведен пример, в котором работают две кнопки: одна – скрывает элемент с `id="test"`, вторая – отображает этот элемент (используется метод `show()`). Обратите внимание, чтобы распараллелить код по двум кнопкам, для каждой из них тоже были определены `id`, по которым определяется какую функцию запускать.

```
<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js">
</script>

<script>
$(document).ready(function () {
    $("#but1").click(function () {
        $("#test").hide();
    });
    $("#but2").click(function () {
        $("#test").show();
    });
});
</script>
</head>

<body>
<h2>This is a heading 2</h2>
<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>
<button id="but1">Click me</button>
<button id="but2">Click me for show</button>
</body>

</html>
```

Рисунок 31.4 – Работа метода `hide()`

Селектор jQuery `class` находит элементы определенного класса. Для поиска элементов определенного класса указывается перед названием точка, например: `$(".test")`, рисунок 31.5

Синтаксис	Описание
<code>\$("*")</code>	Selects all elements
<code>\$(this)</code>	Selects the current HTML element
<code>\$("p.intro")</code>	Selects all <code><p></code> elements with <code>class="intro"</code>
<code>\$("p:first")</code>	Selects the first <code><p></code> element
<code>\$("ul li:first")</code>	Selects the first <code></code> element of the first <code></code>
<code>\$("ul li:first-child")</code>	Selects the first <code></code> element of every <code></code>
<code>\$("[href]")</code>	Selects all elements with an href attribute
<code>\$("a[target='_blank']")</code>	Selects all <code><a></code> elements with a target attribute value equal to <code>"_blank"</code>
<code>\$("a[target!='_blank']")</code>	Selects all <code><a></code> elements with a target attribute value NOT equal to <code>"_blank"</code>
<code>\$(":button")</code>	Selects all <code><button></code> elements and <code><input></code> elements of <code>type="button"</code>
<code>\$("tr:even")</code>	Selects all even <code><tr></code> elements
<code>\$("tr:odd")</code>	Selects all odd <code><tr></code> elements

Рисунок 31.5 – Селекторы jQuery

В таблице 31.1 представлены общие с технологией DOM события. Таблица 31.1 – События jQuery

События мыши	События клавиатуры	События формы	События документа/окна
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

В jQuery, большинство DOM-событий имеют эквивалентный jQuery-метод. Чтобы назначить событие нажатия мышкой на все элементы `<p>` на странице, вы можете написать:

```
$("p").click();
```

Следующий шаг – это определение того, что будет происходить, когда наступит указанное событие. Вы должны определить функцию для события:

```
$("p").click(function(){
    // описание действий в функции
});
```

Часто используемые методы JQuery

`$(document).ready()` метод позволяет вам выполнить функцию, когда документ полностью загружен.

`click()` Эта функция выполняется, когда пользователь нажимает на HTML элемент.

`dblclick()` Срабатывает, когда пользователь двойным щелчком нажимает на HTML-элемент.

`mouseenter()` Выполняется, когда указатель мыши наводится на HTML-элемент.

`blur()` Выполняется, когда поле формы теряет фокус.

`hide()` и `show()` С jQuery вы можете скрывать и отражать HTML-элементы. Также возможно настроить время затухания и появления. Синтаксис:

```
$(selector).hide(speed);
```

```
$(selector).show(speed);
```

Необязательный параметр скорости определяет скорость скрывания / показа, и может принимать следующие значения: "slow", "fast" или в миллисекундах.

5. Порядок выполнения работы

1. Выполнить подключение библиотеки jQuery. Повторить пример из теории, проверить работу метода `hide()`.

2. В исходный файл `Index31.html` добавить в начало две кнопки: одну-для скрывания элементов, другую – для отображения скрытых элементов. Настроить методы `click()` кнопок, так чтобы они то скрывали, то отображали нечетные элементы с классом `MsoNormal`.

3. В исходный файл `Index31.html` добавить в начало две кнопки: одну-для скрывания элементов, другую – для отображения скрытых элементов. Настроить методы `click()` кнопок, так чтобы они то скрывали, то отображали четные элементы типа `<tr>`.

4. В исходный файл `Index31.html` добавить в начало две кнопки: одну-для скрывания элементов, другую – для отображения скрытых элементов. Настроить методы `click()` кнопок, так чтобы они то скрывали, то отображали элементы с атрибутом `href`.

5. В исходный файл `Index31.html` добавить в начало две кнопки: одну-для скрывания элементов, другую – для отображения скрытых элементов. Настроить методы `click()` кнопок, так чтобы они то скрывали, то отображали элементы, у которых атрибут `align` равен значению `center`.

6. В исходный файл `Index31.html` добавить однострочное поле для ввода текстовой информации и кнопку. При вводе в текстовое поле текста кнопка должна скрываться, при потере фокуса однострочного поля кнопка должна показываться.

7. В исходный файл Index31.html добавить кнопку «Тест» и изображение. При двойном клике скрывать изображение, при одном клике – показывать.

6. Форма отчета о работе

Лабораторная работа № ____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы и задания

1. Для чего предназначена библиотека jQuery?
2. Опишите возможные способы подключения библиотеки jQuery к HTML
3. Каким образом можно обратиться к селекторам CSS с помощью библиотеки jQuery?

8. Рекомендуемая литература

1. **JAVASCRIPT.RU** [Электронный ресурс] / Современный учебник JavaScript – 2007—2020 Илья Кантор. – Режим доступа: <https://learn.javascript.ru>. – Дата доступа: 04.03.2020.
2. **Макфарланд, Д.** JavaScript и jQuery. Исчерпывающее руководство / Д. Макфарланд - М. : «Эксмо», 2016 – 880с
3. **Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. 4-е изд. – СПб.: Питер, 2018.
4. **Симпсон, К.** ES6 и не только / К. Симпсон. – СПб.: Питер, 2017.
5. **Хавербеке, М.** Выразительный JavaScript. Современное веб-программирование / М. Хавербеке – СПб.: Питер, 2019.