6 БИБИЛИОТЕКА JQUERY. РАЗРАБОТКА ГИБКИХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

**6.1** **Теоретические сведения**

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.

Есть два варианта подключения:

1) локальная загрузка;

2) удаленная загрузка.

По первому варианту скачиваем библиотеку на жесткий диск и указываем в коде документа путь доступа к ней. Библиотеку jQuery можно скачать с сайта http://jquery.com. По второму варианту библиотека будет подгружаться с сервера при загрузке страницы, для этого необходимо вставить ссылки в документ известные интернет-адреса. В условиях лаборатории применяем первый вариант. Пример подключения:

|  |
| --- |
| !DOCTYPE html>  <html>  <head>  <script src="C:\jq\jquery-1.11.1.js">  </script>  </head>  <body>  </body>  </html> |

jQuery синтаксис

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:

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

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

$(document).ready(function(){

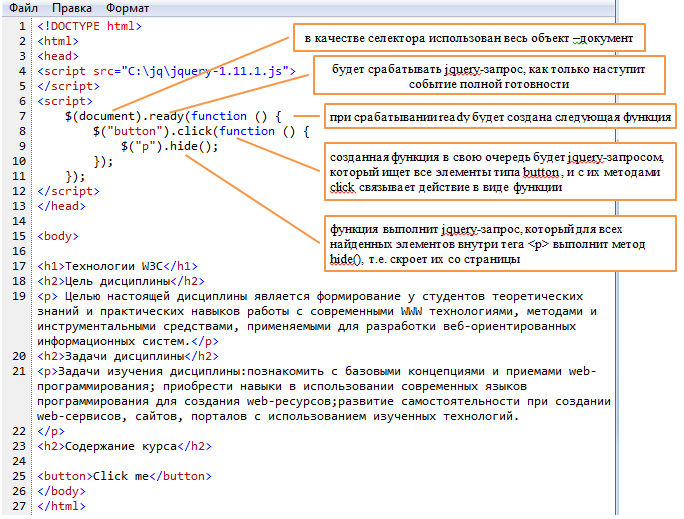
// jQuery-методы размещаем здесь...

});

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 для работы с ним.

jQuery селекторы. jQuery селекторы позволяют вам делать выборку (поиск) и манипулировать с элементами HTML. Селекторы по сути это набор условных обозначений и правил для выборки и манипулирования. 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( ) ):

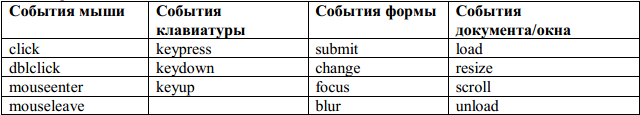


Обратите внимание на то, что селектор в $( )указывается в кавычках, а не в < >.

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.

jQuery-методы событий

Ниже представлены общие с технологией DOM события:



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( ) 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() скрывает и отражает HTML-элементы. Также возможно настроить время затухания и появления.

Синтаксис:

$(selector).hide(speed);

$(selector).show(speed);

или

$(selector).hide(speed,callback);

$(selector).show(speed,callback);

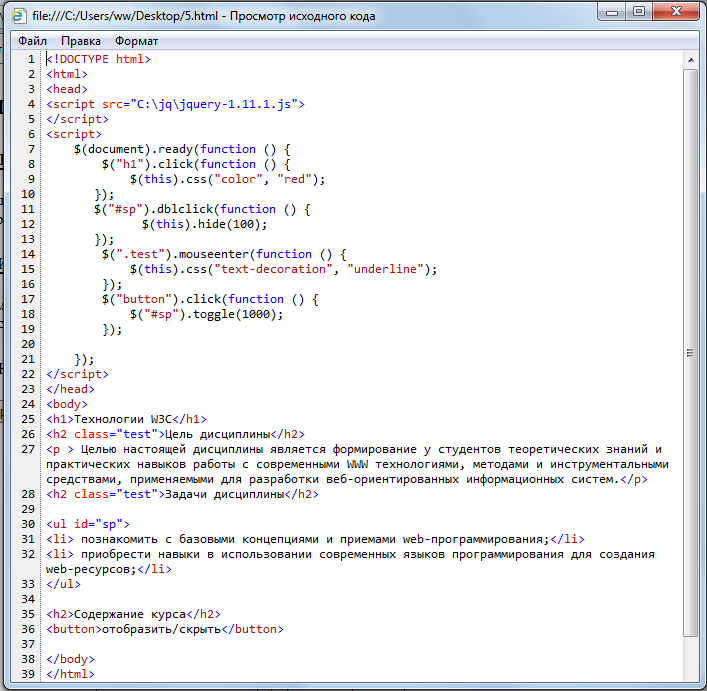
Необязательный параметр скорости определяет скорость скрытия / показа, и может принимать следующие значения: "slow", "fast" или в миллисекундах.

toggle() переключает выполнение методов hide() и show(). Также можно настроить и скорость срабатывания.

Синтаксис:

$(selector).toggle(speed);

В приведенном ниже примере происходят следующие события: заголовок первого уровня при нажатии на него мышкой меняет цвет; список, при двойном щелчке мыши на нем, скрывается со скоростью 100 миллисекунд; элементы класса test при наведении курсора мыши подчеркиваются; при нажатии на кнопку отображается или скрывается список.



Перебор результатов поиска. Результатом поиска является jQuery-объект. Он похож на массив: в нём есть нумерованные элементы и length (длина массива), но методы у него совсем другие. jQuery-объект также называют «jQuery-коллекцией», «элементами, обёрнутыми в jQuery» и пр.

Получить доступ к элементу коллекции можно, указав его индекс в квадратных скобках. Нумерация начинается с нуля.

Для более удобного перебора у jQuery-коллекции есть метод each. Его синтаксис похож на forEach массива:

.each( function(index, item) )

Он выполняет для каждого элемента коллекции перед точкой функцию-аргумент, и передаёт ей номер index и очередной элемент item.

Например:

<!DOCTYPE HTML>

<html>

<body>

<script src="http://code.jquery.com/jquery.js"></script>

<a href="http://wikipedia.ru">Википедия</a>

<ul>

<li><a href="http://jquery.com">jQuery</a></li>

<li><a href="http://sizzlejs.com">Sizzle</a></li>

<li><a href="http://blog.jquery.com">jQuery Blog</li>

</ul>

<script>

var links = $('li a'); // найти все ссылки на странице внутри LI

links.each(function (i, a) {

alert(i + ': ' + a.innerHTML);

if (i == 1) return false; // стоп на элементе коллекции с индексом 1

});

</script>

</body>

</html>

Три основных, но полных jQuery метода для манипулирования данными технологии DOM:

* text() - Задает или возвращает текстовое содержимое выбранных элементов.
* html() - Задает или возвращает содержание отдельных элементов (включая HTML markup).
* val() - Задает или возвращает значение поля формы.

Получение значений атрибутов с помощью attr(). В следующем примере показано, как получить значение атрибута HREF из ссылки:

<!DOCTYPE html>

<html>

<head>

<script src="http://code.jquery.com/jquery.js"></script>

<script>

$(document).ready(function () {

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

alert($("#w3s").attr("href"));

});

});

</script>

</head>

<body>

<p><a href="http://www.w3schools.com" id="w3s">W3Schools.com</a></p>

<button>Show href Value</button>

</body>

</html>

**Получение конкретного элемента.** Даже если элемент найден только один, всё равно результатом будет jQuery-коллекция. Для получения одного элемента из jQuery-коллекции есть несколько способов:

1) Метод get(индекс), работает так же, как прямой доступ:

alert( $('body').get(0) ); // BODY

Если элемента с таким номером нет — вызов get возвратит undefined.

2) Метод eq(индекс) возвращает коллекцию из одного элемента — с данным номером. Он отличается от метода get(индекс) и прямого обращения по индексу тем, что возвращает именно jQuery-коллекцию с одним элементом, а не сам элемент.

// DOM-элемент для первой ссылки

$('a').get(0);

// jQuery-объект из одного элемента: первой ссылки

$('a').eq(0);

Во втором случае вызов eq создаёт новую jQuery-коллекцию, добавляет в нее нулевой элемент и возвращает. Это удобно, если мы хотим дальше работать с этим элементом, используя методы jQuery. Если элемента с таким номером нет — eq возвратит пустую коллекцию.

**6.2 Задание на лабораторную работу**

Продолжаем проект. HTML документ должен содержать обращение к методам библиотеки JQuery: методы должны быть ориентированы на элементы, выбранные селекторами (в том числе с фильтрами); методы, которые используются для CSS-стилей; методы для работы с html; методы по обработке событий; методы визуальных эффектов. Использовать плагины JQuery.