

## Лабораторная работа №6

### Основы jQuery

**Цель:** научиться использовать библиотеку jQuery.

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.

### Начало работы с библиотекой

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:  
<https://developers.google.com/speed/libraries/devguide?hl=ru#jquery> можно загрузить любую из не слишком старых версий. 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 (**!!!** но использовать один из вариантов подключения библиотеки):

```
<!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>
```

## 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:

`$(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"`.

## Событие Ready у объекта страницы Document

Вы, заметите, что 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

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 (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()`):

```
<!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>

```

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

**Задание 1:** Выполнить пример, проверить работу метода `hide()`. Попробовать заменить библиотеку `jquery-latest.js` на другие две, описанные выше, сравнить результаты работы.

### 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 #id

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")`

**В примере** ниже, при нажатии на кнопку ищется элемент с идентификатором test и скрывается со страницы.

```
<!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>
```

**Задание 2:** Выполнить пример, проанализировать отличие его от примера в задании 1.

Ниже 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>
```

**Задание3:** Выполнить пример, проанализировать отличие его от примера в задании 2.

### **.class селектор**

Селектор jQuery class находит элементы определенного класса.

Для поиска элементов определенного класса указывается перед названием точка, например:

`$(".test")`

**В примере ниже при нажатии на кнопку элементы с классом="test" будут скрыты:**

```
<!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 class="test">This is a heading</h2>
<p class="test">This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

**Задание 4:** Выполнить пример, проанализировать отличие его от примера в задании 3.

### **Другие примеры jQuery селекторов (делать необязательно, просто при необходимости для ознакомления)**

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

### **Индивидуальное задание 1:**

**Вариант 1:** в исходный файл Experiments.html добавить в начало две кнопки: одну-для скрытия элементов, другую – для отображения скрытых элементов. Настроить методы click( ) кнопок, так чтобы они то скрывали, то отображали нечетные элементы с классом `MsoNormal`.

**Вариант 2:** в исходный файл Experiments.html добавить в начало две кнопки: одну-для скрытия элементов, другую – для отображения скрытых элементов. Настроить методы click( ) кнопок, так чтобы они то скрывали, то отображали четные элементы типа `<tr>`.

**Вариант 3:** в исходный файл Experiments.html добавить в начало две кнопки: одну-для скрытия элементов, другую – для отображения скрытых элементов. Настроить методы click( ) кнопок, так чтобы они то скрывали, то отображали элементы с атрибутом `href`.

**Вариант 4:** в исходный файл Experiments.html добавить в начало две кнопки: одну-для скрытия 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`.

### jQuery-методы событий

Ниже представлены общие с технологией DOM события:

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

### jQuery-синтаксис для методов событий

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()`**

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

**`click()`**

Эта функция выполняется, когда пользователь нажимает на HTML элемент.

**В примере** событие `click()` активируется на `<p>` элементах, скрывая текущий `<p>` элемент:

```
<!DOCTYPE html>  
<html>  
<head>  
<script src="http://code.jquery.com/jquery-latest.js">  
</script>  
  
<script>  
    $(document).ready(function () {  
        $("p").click(function () {
```

```

        $(this).hide();
    });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>
</body>
</html>

```

**Задание 5:** Выполнить пример, проверить работу.

**Обратите внимание!** `$(this).hide();` //указатель `this` позволяет выполнять действие //с текущим активированным элементом

### **dblclick()**

Срабатывает, когда пользователь двойным щелчком нажимает на HTML-элемент:

```

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

<script>
    $(document).ready(function () {
        $("p").dblclick(function () {
            $(this).hide();
        });
    });
</script>
</head>
<body>

<p>If you double-click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>

```

**Задание 6:** Выполнить пример, проверить работу.

### **mouseenter()**

Выполняется, когда указатель мыши наводится на HTML-элемент:

```

<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js">
</script>
<script>
    $(document).ready(function () {
        $("#p1").mouseenter(function () {
            alert("You entered p1!");
        });
    });
</script>
</head>
<body>

<p id="p1">Enter this paragraph.</p>

```

```
</body>
</html>
```

**Задание 7:** Выполнить пример, проверить работу.

### blur()

Выполняется, когда поле формы теряет фокус.

```
<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js">
</script>
<script>
    $(document).ready(function () {
        $("input").focus(function () {
            $(this).css("background-color", "#cccccc");
        });
        $("input").blur(function () {
            $(this).css("background-color", "#ffffff");
        });
    });
</script>
</head>
<body>

Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">

</body>
</html>
```

**Задание 8:** Выполнить пример, проверить работу. Проверить как работает focus() и blur(), а также разобраться с настройкой стилей через .css("свойство", "значение").

### 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);`

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.

Следующий пример демонстрирует параметр скорости скрытия элемента:

```
<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js">
</script>
<script>
    $(document).ready(function () {
        $("button").click(function () {
            $("p").hide(1000);
        });
    });
</script>
```



```

</script>
</head>
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>
</body>
</html>

```

**Задание 9:** Выполнить пример, проверить работу.

### toggle()

С помощью метода можно переключать выполнение методов `hide()` и `show()`. Также можно настроить и скорость срабатывания.

**Синтаксис:**

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

```

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

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

<button>Toggle</button>
<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>
</body>
</html>

```

**Задание 10:** Выполнить пример, проверить работу.

### Индивидуальное задание 2:

**Вариант 1:** в исходный файл Experiments2.html добавить обработку события «наведение мышки на элемент». Если этот элемент относится к таблице (в теге `<img>`), то при наведении на него курсора, он должен скрываться.

**Вариант 2:** в исходный файл Experiments2.html добавить обработку события «двойной щелчок мышки по элементу». Для элемента с атрибутом `align="center"` нужно настроить исчезновение/появление элемента.

**Вариант 3:** в исходный файл Experiments2.html добавить обработку события «изменение размера окна». При срабатывании этого метода скрыть с документа все элементы со ссылками (`href`).

**Вариант 4:** в исходный файл Experiments2.html добавить элемент формы «поле ввода», настроить для него обработку `keypress`, а именно случайным образом менять цвет этого поля (свойство `background-color`).

### Fading методы

С JQuery вы можете исключать и включать видимость элементов.

Набор методов:

- `fadeIn()`

- fadeOut()
- fadeToggle()
- fadeTo()

### fadeIn()

fadeIn() используется, чтобы реализовать проявление в скрытых элементах (например, элемент <div> скрытый).

Можно также использовать настройку скорости срабатывания.

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

<script>
$(document).ready(function () {
    $("button").click(function () {
        $("#div1").fadeIn();
        $("#div2").fadeIn("slow");
        $("#div3").fadeIn(3000);
    });
});
</script>
</head>

<body>
<p>Demonstrate fadeIn() with different parameters.</p>
<button>Click to fade in boxes</button>
<br><br>
<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>

</body>
</html>
```

**Задание 11:** Выполнить пример, проверить работу. Разобраться с настройками элементов <div>, обратите внимание на свойство `display:none`, которое при загрузке страницы позволяет не отражать <div>.

### fadeOut()

fadeOut() используется, чтобы реализовать скрытие элементов из зоны видимости.

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

<script>
$(document).ready(function () {
    $("button").click(function () {
        $("#div1").fadeOut();
        $("#div2").fadeOut("slow");
        $("#div3").fadeOut(3000);
    });
});
</script>
</head>

<body>
<p>Demonstrate fadeOut() with different parameters.</p>
```

```

<button>Click to fade out boxes</button>
<br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>
</html>

```

**Задание 12:** Выполнить пример, проверить работу. Обратите внимание на отсутствие свойства `display:none`, поэтому настройки элементов `<div>` сначала отражаются.

### Индивидуальное задание 3:

**Вариант 1:** создать файл Individual\_3.html, добавить два рисунка, невидимые при загрузке страницы, и две кнопки. При нажатии на одну кнопку должно в цикле срабатывать проявление первой картинке и исчезновение второй картинке со скоростью 2000, а затем наоборот, при нажатии на вторую кнопку обе картинке должны появиться и прекратить мигание.

**Вариант 2:** создать файл Individual\_3.html, добавить пять областей div с id "div1", "div2", ..., "div5", в которые поместить тексты, невидимые при загрузке страницы, и две кнопки. При нажатии на одну кнопку должно определяться случайное число от 1 до 5. В зависимости от этого числа должна проявиться соответствующая область div со скоростью 3000. При нажатии на вторую кнопку все области div снова скрываются.

**Вариант 3:** создать файл Individual\_3.html, добавить 10 параграфов (`<p>`) с каким-либо текстом, невидимые при загрузке страницы, и кнопку. При нажатии на кнопку должно в цикле срабатывать проявление четных параграфов и исчезновение нечетных со скоростью 2000, а затем наоборот.

**Вариант 4:** создать файл Individual\_3.html, добавить пять областей div 40x40 пикселей с id "div1", "div2", ..., "div5", раскрашенные в разные цвета, невидимые при загрузке страницы, и кнопку. При нажатии на кнопку должно в цикле срабатывать волнообразное проявление областей div слева-направо, а затем наоборот, исчезновение. Цикл работает до тех пор, пока не закрыта страница.

## Sliding методы

Реализацию движения по принципу слайдов можно выполнять с помощью методов:

- `slideDown()`
- `slideUp()`
- `slideToggle()`

### slideDown()

Используется для скольжения элемента вниз.

#### Синтаксис:

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

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

```

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

<script>
    $(document).ready(function () {
        $("#flip").click(function () {
            $("#panel").slideDown("slow");
        });
    });

```

```

</script>

<style type="text/css">
#panel,#flip
{
padding:5px;
text-align:center;
background-color:#e5eccc;
border:solid 1px #c3c3c3;
}
#panel
{
padding:50px;
display:none;
}
</style>
</head>
<body>

<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>

</body>
</html>

```

**Задание 13:** Выполнить пример, проверить работу. Разобраться с элементом `<style type="text/css">`. Обратите внимание, что настройки делаются как общие для нескольких элементов, так и индивидуальные.

### Метод `animate()`

`Animate()` используется для создания пользовательской анимации.

#### Синтаксис:

`$(selector).animate({params},speed,callback);`

Обязательный **params** определяется параметрами CSS-свойств для анимации.

Необязательный параметр **speed** определяет длительность эффекта. Он может принимать следующие значения: "slow", "fast", или в миллисекундах.

В следующем примере показан простой вариант использования метода, он перемещает элемент `<div>` влево, пока он не достиг значения свойства `left=250px`:

```

<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery-latest.js">
</script>
<script>
    $(document).ready(function () {
        $("button").click(function () {
            $("div").animate({ left: '250px' });
        });
    });
</script>
</head>
<body>
<button>Start Animation</button>
<p>By default, all HTML elements have a static position, and cannot be moved. To
manipulate the position, remember to first set the CSS position property of the element
to relative, fixed, or absolute!</p>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;">
</div>
</body>
</html>

```

**Задание 14:** Выполнить пример, проверить работу. Разобраться с элементом `<div>`.

С прочими примерами анимации можно познакомиться на [http://www.w3schools.com/jquery/jquery\\_animate.asp](http://www.w3schools.com/jquery/jquery_animate.asp)

## 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. Используем jQuery, чтобы выбрать все элементы по селектору `li a` и перебрать их:

```
<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery.js"></script>
<script>
    $(document).ready(function () {
        $("button").click(function () {
            var links = $('li a');
            // перебор результатов
            for (var i = 0; i < links.length; i++) {
                alert(links[i].href);
            }
        });
    });
</script>
</head>

<body>
<ul>
    <li><a href="http://jquery.com">jQuery</a></li>
    <li><a href="http://jqueryui.com">jQuery UI</a></li>
    <li><a href="http://blog.jquery.com">jQuery Blog</a></li>
</ul>
<button>Start query</button>

</body>
</html>
```

**Задание 15:** Выполнить пример, проверить работу. Разобраться, как организована работа с переменной **links** и работа с ней в цикле.

## Контекст поиска

А что, если контекст поиска содержит много элементов? Например, как будет работать запрос `$( 'a', 'li' )`, если `<li>` в документе много?

Если в контексте много элементов, то поиск будет произведён в каждом из них. А затем результаты будут объединены.

Повторы элементов при этом отфильтровываются, то есть два раза один и тот же элемент в результате не появится.

Например, найдём `$( 'a', 'li' )` в многоуровневом списке:

```
<!DOCTYPE HTML>
<html>
<body>
<script src="http://code.jquery.com/jquery.js"></script>

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

```

</ul>

<script>

    var links = $('a', 'li');
    for (var i = 0; i < links.length; i++) {
        alert(i + ": " + links[i].href); // 3 ссылки по очереди
    }
</script>

</body>
</html>

```

**Задание 16:** Выполнить пример, проверить работу. Разобраться, как организована работа jQuery, когда скрипт размещается в теле страницы (нет объявления функций и привязок к методам).

!!!Обратите внимание, что здесь jQuery-запрос идет в теле документа без привязки к событию, поэтому работает в режиме немедленного запуска (а не срабатывания по принципу триггера на событие в загруженном документе).

## Метод each()

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`.

Используем его вместо `for`, чтобы перебрать коллекцию найденных ссылок:

```

$('li a').each(function(i, a) {
    alert( i + ": " + a.href);
});

```

У `.each` есть важная возможность, которой нет в `forEach`: возврат `false` из функции прекращает перебор.

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>

```

**Задание 17:** Выполнить пример, проверить работу. Разобраться, как организована работа метода `each()`.

## Получение контента с помощью text(), html(), and val()

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

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

В следующем примере показано, как получить контент с помощью text() и html() методов:

```
<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery.js"></script>
<script>
    $(document).ready(function () {
        $("#btn1").click(function () {
            alert("Text: " + $("#test").text());
        });
        $("#btn2").click(function () {
            alert("HTML: " + $("#test").html());
        });
    });
</script>
</head>

<body>
<p id="test">This is some <b>bold</b> text in a paragraph.</p>
<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>
</body>
</html>
```

**Задание 18:** Выполнить пример, проверить работу. Разобраться, как организована работа метода text() и html().

В следующем примере показано, как получить значение поля ввода с помощью метода VAL():

```
<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery.js"></script>
<script>
    $(document).ready(function () {
        $("button").click(function () {
            alert("Value: " + $("#test").val());
        });
    });
</script>
</head>

<body>
<p>Name: <input type="text" id="test" value="Mickey Mouse"></p>
<button>Show Value</button>
</body>
</html>
```

**Задание 19:** Выполнить пример, проверить работу. Разобраться, как организована работа метода val().

## 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>

```

**Задание 20:** Выполнить пример, проверить работу. Разобраться, как организована работа метода attr(). Определите, значения каких еще атрибутов можно так получить.

#### Индивидуальное задание 4:

**Вариант 1:** создать файл Individual\_4.html, добавить сверху кнопку, а затем пять параграфов с каким-либо текстом и между каждым параграфом по маленькому рисунку. При нажатии на кнопку среди всех найденных параграфов для 3-го и 5-го изменить положение – сместить его на 100 пт влево относительно текущего положения. При нажатии на любую из картинок должно происходить движение текущей вниз на 30 пт.

**Вариант 2:** создать файл Individual\_4.html, добавить сверху кнопку, а затем 6 маленьких рисунков одинакового размера, один под другим. При нажатии на кнопку среди всех найденных рисунков для четных выполнить эффект слайд-шоу – заезд/скрытие рисунка под соответствующим ему верхним рисунком со скоростью «slow».

**Вариант 3:** создать файл Individual\_4.html, добавить сверху кнопку, а затем 6 параграфов с каким-либо текстом, один под другим. При нажатии на кнопку среди всех найденных рисунков для нечетных выполнить извлечение текста и присвоение этого текста соответствующему нижнему/четному параграфу и сместить нечетные элементы на 40 пт влево.

**Вариант 4:** создать файл Individual\_4.html, добавить сверху кнопку, а затем 6 маленьких рисунков одинакового размера, один под другим. При нажатии на кнопку среди всех найденных рисунков для нечетных выполнить эффект слайд-шоу – заезд/скрытие рисунка под соответствующим ему нижним рисунком со скоростью «slow».

### 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.

### **Отчет по лабораторной работе**

В соответствии со структурой заготовки отчета и примером оформления оформить в отчете все задания, выполняемые в ходе лабораторной работы, а также индивидуальные задания по вариантам. Файл с отчетом называть по шаблону: **Фамилия\_лаб\_раб\_номер**.

Отчет предоставляется в электронном виде либо лично преподавателю, либо на электронную почту для проверки. Также по результатам лабораторной работы на следующем за ней занятии проводится выборочный опрос по командам языка.

## Приложение А

### Справочные материалы для просмотра примеров

#### jQuery Selectors

Use our excellent [jQuery Selector Tester](#) to experiment with the different selectors.

Selector	Example	Selects
<a href="#"><u>*</u></a>	<code>\$("*")</code>	All elements
<a href="#"><u>#id</u></a>	<code>\$("#lastname")</code>	The element with id="lastname"
<a href="#"><u>.class</u></a>	<code>\$(".intro")</code>	All elements with class="intro"
<a href="#"><u>.class.class</u></a>	<code>\$(".intro.demo")</code>	All elements with the class "intro" or "demo"
<a href="#"><u>element</u></a>	<code>\$("p")</code>	All <p> elements
<a href="#"><u>el1,el2,el3</u></a>	<code>\$("h1,div,p")</code>	All <h1>, <div> and <p> elements
<a href="#"><u>:first</u></a>	<code>\$("p:first")</code>	The first <p> element
<a href="#"><u>:last</u></a>	<code>\$("p:last")</code>	The last <p> element
<a href="#"><u>:even</u></a>	<code>\$("tr:even")</code>	All even <tr> elements
<a href="#"><u>:odd</u></a>	<code>\$("tr:odd")</code>	All odd <tr> elements
<a href="#"><u>:first-child</u></a>	<code>\$("p:first-child")</code>	All <p> elements that are the first child of their parent
<a href="#"><u>:first-of-type</u></a>	<code>\$("p:first-of-type")</code>	All <p> elements that are the first <p> element of their parent
<a href="#"><u>:last-child</u></a>	<code>\$("p:last-child")</code>	All <p> elements that are the last child of their parent
<a href="#"><u>:last-of-type</u></a>	<code>\$("p:last-of-type")</code>	All <p> elements that are the last <p> element of their parent
<a href="#"><u>:nth-child(n)</u></a>	<code>\$("p:nth-child(2)")</code>	All <p> elements that are the 2nd child of their parent
<a href="#"><u>:nth-last-child(n)</u></a>	<code>\$("p:nth-last-child(2)")</code>	All <p> elements that are the 2nd child of their parent, counting from the last child
<a href="#"><u>:nth-of-type(n)</u></a>	<code>\$("p:nth-of-type(2)")</code>	All <p> elements that are the 2nd <p> element of their parent
<a href="#"><u>:nth-last-of-type(n)</u></a>	<code>\$("p:nth-last-of-type(2)")</code>	All <p> elements that are the 2nd <p> element of their parent, counting from the last child
<a href="#"><u>:only-child</u></a>	<code>\$("p:only-child")</code>	All <p> elements that are the only child of their parent
<a href="#"><u>:only-of-type</u></a>	<code>\$("p:only-of-type")</code>	All <p> elements that are the only child, of its type, of their parent
<a href="#"><u>parent &gt; child</u></a>	<code>\$("div &gt; p")</code>	All <p> elements that are a direct child of a <div> element
<a href="#"><u>parent descendant</u></a>	<code>\$("div p")</code>	All <p> elements that are descendants of a <div> element
<a href="#"><u>element + next</u></a>	<code>\$("div + p")</code>	The <p> element that are next to each <div> elements
<a href="#"><u>element ~ siblings</u></a>	<code>\$("div ~ p")</code>	All <p> elements that are siblings of a <div> element
<a href="#"><u>:eq(index)</u></a>	<code>\$("ul li:eq(3)")</code>	The fourth element in a list (index starts at 0)
<a href="#"><u>:gt(no)</u></a>	<code>\$("ul li:gt(3)")</code>	List elements with an index greater than 3
<a href="#"><u>:lt(no)</u></a>	<code>\$("ul li:lt(3)")</code>	List elements with an index less than 3
<a href="#"><u>:not(selector)</u></a>	<code>\$("input:not(:empty)")</code>	All input elements that are not empty
<a href="#"><u>:header</u></a>	<code>\$(":header")</code>	All header elements <h1>, <h2> ...
<a href="#"><u>:animated</u></a>	<code>\$(":animated")</code>	All animated elements
<a href="#"><u>:focus</u></a>	<code>\$(":focus")</code>	The element that currently has focus
<a href="#"><u>:contains(text)</u></a>	<code>\$(":contains('Hello')")</code>	All elements which contains the text "Hello"
<a href="#"><u>:has(selector)</u></a>	<code>\$("div:has(p)")</code>	All <div> elements that have a <p> element
<a href="#"><u>:empty</u></a>	<code>\$(":empty")</code>	All elements that are empty
<a href="#"><u>:parent</u></a>	<code>\$(":parent")</code>	All elements that are a parent of another element
<a href="#"><u>:hidden</u></a>	<code>\$("p:hidden")</code>	All hidden <p> elements
<a href="#"><u>:visible</u></a>	<code>\$("table:visible")</code>	All visible tables
<a href="#"><u>:root</u></a>	<code>\$(":root")</code>	The document's root element
<a href="#"><u>:lang(language)</u></a>	<code>\$("p:lang(de)")</code>	All <p> elements with a lang attribute value starting with "de"
<a href="#"><u>[attribute]</u></a>	<code>\$("[href]")</code>	All elements with a href attribute
<a href="#"><u>[attribute=value]</u></a>	<code>\$("[href='default.htm']")</code>	All elements with a href attribute value equal to "default.htm"
<a href="#"><u>[attribute!=value]</u></a>	<code>\$("[href!='default.htm']")</code>	All elements with a href attribute value not equal to "default.htm"
<a href="#"><u>[attribute\$=value]</u></a>	<code>\$("[href\$='.jpg']")</code>	All elements with a href attribute value ending with ".jpg"

<a href="#">[attribute =value]</a>	<code>\$("[hreflang =en"])</code>	All elements with a hreflang attribute value starting with "en"
<a href="#">[attribute^=value]</a>	<code>\$("[name^=hello"])</code>	All elements with a name attribute value starting with "hello"
<a href="#">[attribute~=value]</a>	<code>\$("[name~=hello"])</code>	All elements with a name attribute value containing the word "hello"
<a href="#">[attribute*=value]</a>	<code>\$("[name*=hello"])</code>	All elements with a name attribute value containing the string "hello"
<a href="#">:input</a>	<code>\$(":input")</code>	All input elements
<a href="#">:text</a>	<code>\$(":text")</code>	All input elements with type="text"
<a href="#">:password</a>	<code>\$(":password")</code>	All input elements with type="password"
<a href="#">:radio</a>	<code>\$(":radio")</code>	All input elements with type="radio"
<a href="#">:checkbox</a>	<code>\$(":checkbox")</code>	All input elements with type="checkbox"
<a href="#">:submit</a>	<code>\$(":submit")</code>	All input elements with type="submit"
<a href="#">:reset</a>	<code>\$(":reset")</code>	All input elements with type="reset"
<a href="#">:button</a>	<code>\$(":button")</code>	All input elements with type="button"
<a href="#">:image</a>	<code>\$(":image")</code>	All input elements with type="image"
<a href="#">:file</a>	<code>\$(":file")</code>	All input elements with type="file"
<a href="#">:enabled</a>	<code>\$(":enabled")</code>	All enabled input elements
<a href="#">:disabled</a>	<code>\$(":disabled")</code>	All disabled input elements
<a href="#">:selected</a>	<code>\$(":selected")</code>	All selected input elements
<a href="#">:checked</a>	<code>\$(":checked")</code>	All checked input elements

### *jQuery Event Methods*

Event methods trigger or attach a function to an event handler for the selected elements.

The following table lists all the jQuery methods used to handle events.

Method	Description
<a href="#">bind()</a>	Attaches event handlers to elements
<a href="#">blur()</a>	Attaches/Triggers the blur event
<a href="#">change()</a>	Attaches/Triggers the change event
<a href="#">click()</a>	Attaches/Triggers the click event
<a href="#">dblclick()</a>	Attaches/Triggers the double click event
<a href="#">delegate()</a>	Attaches a handler to current, or future, specified child elements of the matching elements
<a href="#">die()</a>	Removed in version 1.9. Removes all event handlers added with the live() method
<a href="#">error()</a>	Attaches/Triggers the error event
<a href="#">event.currentTarget</a>	The current DOM element within the event bubbling phase
<a href="#">event.data</a>	Contains the optional data passed to an event method when the current executing handler is bound
<a href="#">event.delegateTarget</a>	Returns the element where the currently-called jQuery event handler was attached
<a href="#">event.isDefaultPrevented()</a>	Returns whether event.preventDefault() was called for the event object
<a href="#">event.isImmediatePropagationStopped()</a>	Returns whether event.stopImmediatePropagation() was called for the event object
<a href="#">event.isPropagationStopped()</a>	Returns whether event.stopPropagation() was called for the event object
<a href="#">event.namespace</a>	Returns the namespace specified when the event was triggered
<a href="#">event.pageX</a>	Returns the mouse position relative to the left edge of the document
<a href="#">event.pageY</a>	Returns the mouse position relative to the top edge of the document
<a href="#">event.preventDefault()</a>	Prevents the default action of the event
<a href="#">event.relatedTarget</a>	Returns which element being entered or exited on mouse movement.
<a href="#">event.result</a>	Contains the last/previous value returned by an event handler triggered by the specified event
<a href="#">event.stopImmediatePropagation()</a>	Prevents other event handlers from being called
<a href="#">event.stopPropagation()</a>	Prevents the event from bubbling up the DOM tree, preventing any parent handlers from being notified of the event
<a href="#">event.target</a>	Returns which DOM element triggered the event
<a href="#">event.timeStamp</a>	Returns the number of milliseconds since January 1, 1970, when the event is triggered

<a href="#">event.type</a>	Returns which event type was triggered
<a href="#">event.which</a>	Returns which keyboard key or mouse button was pressed for the event
<a href="#">focus()</a>	Attaches/Triggers the focus event
<a href="#">focusin()</a>	Attaches an event handler to the focusin event
<a href="#">focusout()</a>	Attaches an event handler to the focusout event
<a href="#">hover()</a>	Attaches two event handlers to the hover event
<a href="#">keydown()</a>	Attaches/Triggers the keydown event
<a href="#">keypress()</a>	Attaches/Triggers the keypress event
<a href="#">keyup()</a>	Attaches/Triggers the keyup event
<a href="#">live()</a>	Removed in version 1.9. Adds one or more event handlers to current, or future, selected elements
<a href="#">load()</a>	Attaches an event handler to the load event
<a href="#">mousedown()</a>	Attaches/Triggers the mousedown event
<a href="#">mouseenter()</a>	Attaches/Triggers the mouseenter event
<a href="#">mouseleave()</a>	Attaches/Triggers the mouseleave event
<a href="#">mousemove()</a>	Attaches/Triggers the mousemove event
<a href="#">mouseout()</a>	Attaches/Triggers the mouseout event
<a href="#">mouseover()</a>	Attaches/Triggers the mouseover event
<a href="#">mouseup()</a>	Attaches/Triggers the mouseup event
<a href="#">off()</a>	Removes event handlers attached with the on() method
<a href="#">on()</a>	Attaches event handlers to elements
<a href="#">one()</a>	Adds one or more event handlers to selected elements. This handler can only be triggered once per element
<a href="#">\$.proxy()</a>	Takes an existing function and returns a new one with a particular context
<a href="#">ready()</a>	Specifies a function to execute when the DOM is fully loaded
<a href="#">resize()</a>	Attaches/Triggers the resize event
<a href="#">scroll()</a>	Attaches/Triggers the scroll event
<a href="#">select()</a>	Attaches/Triggers the select event
<a href="#">submit()</a>	Attaches/Triggers the submit event
<a href="#">toggle()</a>	Removed in version 1.9. Attaches two or more functions to toggle between for the click event
<a href="#">trigger()</a>	Triggers all events bound to the selected elements
<a href="#">triggerHandler()</a>	Triggers all functions bound to a specified event for the selected elements
<a href="#">unbind()</a>	Removes an added event handler from selected elements
<a href="#">undelegate()</a>	Removes an event handler to selected elements, now or in the future
<a href="#">unload()</a>	Attaches an event handler to the unload event

### *jQuery Effect Methods*

The following table lists all the jQuery methods for creating animation effects.

Method	Description
<a href="#">animate()</a>	Runs a custom animation on the selected elements
<a href="#">clearQueue()</a>	Removes all remaining queued functions from the selected elements
<a href="#">delay()</a>	Sets a delay for all queued functions on the selected elements
<a href="#">dequeue()</a>	Removes the next function from the queue, and then executes the function
<a href="#">fadeIn()</a>	Fades in the selected elements
<a href="#">fadeOut()</a>	Fades out the selected elements
<a href="#">fadeTo()</a>	Fades in/out the selected elements to a given opacity
<a href="#">fadeToggle()</a>	Toggles between the fadeIn() and fadeOut() methods
<a href="#">finish()</a>	Stops, removes and completes all queued animations for the selected elements
<a href="#">hide()</a>	Hides the selected elements
<a href="#">queue()</a>	Shows the queued functions on the selected elements
<a href="#">show()</a>	Shows the selected elements
<a href="#">slideDown()</a>	Slides-down (shows) the selected elements
<a href="#">slideToggle()</a>	Toggles between the slideUp() and slideDown() methods
<a href="#">slideUp()</a>	Slides-up (hides) the selected elements
<a href="#">stop()</a>	Stops the currently running animation for the selected elements
<a href="#">toggle()</a>	Toggles between the hide() and show() methods

## *jQuery HTML / CSS Methods*

The following table lists all the methods used to manipulate the HTML and CSS.

The methods below work for both HTML and XML documents. Exception: the `html()` method.

Method	Description
<a href="#"><u>addClass()</u></a>	Adds one or more class names to selected elements
<a href="#"><u>after()</u></a>	Inserts content after selected elements
<a href="#"><u>append()</u></a>	Inserts content at the end of selected elements
<a href="#"><u>appendTo()</u></a>	Inserts HTML elements at the end of selected elements
<a href="#"><u>attr()</u></a>	Sets or returns attributes/values of selected elements
<a href="#"><u>before()</u></a>	Inserts content before selected elements
<a href="#"><u>clone()</u></a>	Makes a copy of selected elements
<a href="#"><u>css()</u></a>	Sets or returns one or more style properties for selected elements
<a href="#"><u>detach()</u></a>	Removes selected elements (keeps data and events)
<a href="#"><u>empty()</u></a>	Removes all child nodes and content from selected elements
<a href="#"><u>hasClass()</u></a>	Checks if any of the selected elements have a specified class name
<a href="#"><u>height()</u></a>	Sets or returns the height of selected elements
<a href="#"><u>html()</u></a>	Sets or returns the content of selected elements
<a href="#"><u>innerHeight()</u></a>	Returns the height of an element (includes padding, but not border)
<a href="#"><u>innerWidth()</u></a>	Returns the width of an element (includes padding, but not border)
<a href="#"><u>insertAfter()</u></a>	Inserts HTML elements after selected elements
<a href="#"><u>insertBefore()</u></a>	Inserts HTML elements before selected elements
<a href="#"><u>offset()</u></a>	Sets or returns the offset coordinates for selected elements (relative to the document)
<a href="#"><u>offsetParent()</u></a>	Returns the first positioned parent element
<a href="#"><u>outerHeight()</u></a>	Returns the height of an element (includes padding and border)
<a href="#"><u>outerWidth()</u></a>	Returns the width of an element (includes padding and border)
<a href="#"><u>position()</u></a>	Returns the position (relative to the parent element) of an element
<a href="#"><u>prepend()</u></a>	Inserts content at the beginning of selected elements
<a href="#"><u>prependTo()</u></a>	Inserts HTML elements at the beginning of selected elements
<a href="#"><u>prop()</u></a>	Sets or returns properties/values of selected elements
<a href="#"><u>remove()</u></a>	Removes the selected elements (including data and events)
<a href="#"><u>removeAttr()</u></a>	Removes one or more attributes from selected elements
<a href="#"><u>removeClass()</u></a>	Removes one or more classes from selected elements
<a href="#"><u>removeProp()</u></a>	Removes a property set by the <code>prop()</code> method
<a href="#"><u>replaceAll()</u></a>	Replaces selected elements with new HTML elements
<a href="#"><u>replaceWith()</u></a>	Replaces selected elements with new content
<a href="#"><u>scrollLeft()</u></a>	Sets or returns the horizontal scrollbar position of selected elements
<a href="#"><u>scrollTop()</u></a>	Sets or returns the vertical scrollbar position of selected elements
<a href="#"><u>text()</u></a>	Sets or returns the text content of selected elements
<a href="#"><u>toggleClass()</u></a>	Toggles between adding/removing one or more classes from selected elements
<a href="#"><u>unwrap()</u></a>	Removes the parent element of the selected elements
<a href="#"><u>val()</u></a>	Sets or returns the value attribute of the selected elements (for form elements)
<a href="#"><u>width()</u></a>	Sets or returns the width of selected elements
<a href="#"><u>wrap()</u></a>	Wraps HTML element(s) around each selected element
<a href="#"><u>wrapAll()</u></a>	Wraps HTML element(s) around all selected elements
<a href="#"><u>wrapInner()</u></a>	Wraps HTML element(s) around the content of each selected element