Веб-разработка
Библиотека jQuery

Шумилов Вадим Валерьевич, к.т.н.

Тензор, 2017

JavaScript библиотека jQuery

jQuery

- 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

Установка jQuery

Добавление jQuery на страницы:

```
<script type="text/javascript"
    src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js">
</script>
```

jQuery Селекторы

Селекторы используются для **доступа** к элементам страницы

Синтаксис:

$(селектор)

Знак **$** сообщает, что символы, идущие после него, являются jQuery кодом.

Селектор позволяет выбрать элемент на странице.

А что такое «селектор»??

jQuery Селекторы

Например:

$("p") - будут выбраны все элементы **<p>**, которые находятся на странице

$(".par") - будут выбраны все элементы на странице с **class="par"**

$("#par") - будет выбран первый элемент на странице с **id="par"**

jQuery Селекторы

- **Селекторами** называют строчные выражения, с помощью которых задаются условия поиска элементов DOM на странице

- Строчные выражения пишутся **в стиле CSS**

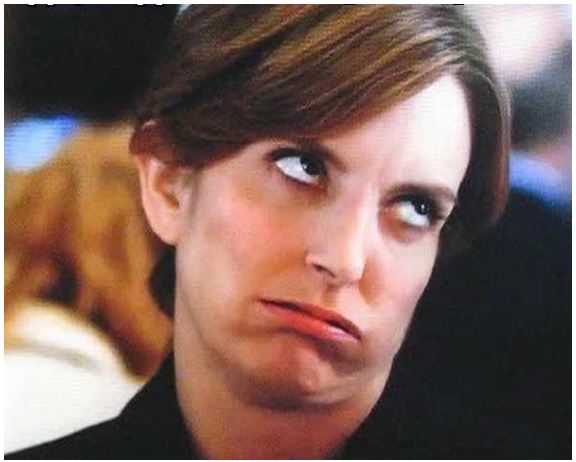
- Селекторы позволяют **находить** элементы по различным **признакам**: значению атрибутов, содержимому элементов, родительским элементам, дочерним элементам, порядковым номерам, именам классов, идентификаторов и тегов

jQuery Селекторы vs функции JavaScript

$("#par") vs document.getElementById("par")

$(".par") vs document.body.getElementsByClassName("par")

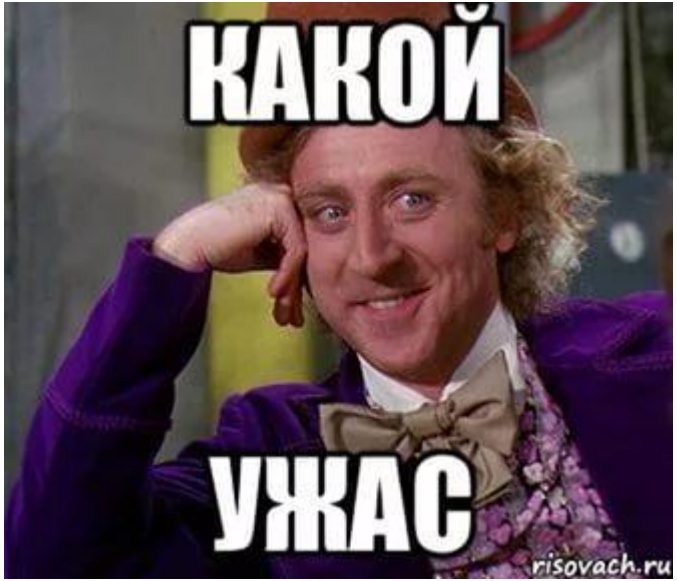
$("p") vs document.body.getElementsByTagName("p")



jQuery Селекторы

Разновидности селекторов:

- Базовые селекторы
- Комбинированные селекторы
- Селекторы по атрибутам
- Простые фильтры
- Фильтры по содержимому
- Фильтры дочерних элементов
- Фильтры элементов форм



jQuery Селекторы

Базовые

Селектор	Возвращает
"*"	все элементы
".className"	элементы с классом className
"#idName"	элемент с идентификатором idName
"tagName"	элементы с заданным именем тега

jQuery Селекторы

Комбинированные селекторы

Селектор	Возвращает
"first, second, ..."	Элементы, удовлетворяющие любому из селекторов first, second, ...
"outer inner"	Элементы из inner, которые являются потомками (т.е. лежат внутри) элементов из outer
"parent > child"	Элементы из child, которые являются прямыми потомками элементов из parent
"prev + next"	Элементы из next, которые следуют непосредственно за элементами из prev
"prev ~ next"	Элементы из next, которые следуют за элементами из prev

jQuery Селекторы

Подробнее про селекторы:

http://jquery.page2page.ru/index.php5/Селекторы

jQuery Команды

Код jQuery состоит из последовательно идущих **команд**.

Стандартный синтаксис jQuery команд:

```
$(селектор).метод();
```

Метод задает **действие**, которое необходимо совершить над выбранным **элементом**.

jQuery Команды

Методы в jQuery разделяются на следующие **группы**:

- Методы для манипулирования DOM;
- Методы для оформления элементов;
- Методы для создания AJAX запросов;
- Методы для создания эффектов;
- Методы для привязки обработчиков событий.

Обработчики событий jQuery

Обработчики событий - это функции, код которых исполняется только после совершения определенных действий.

anonymous function / callback method

```
function()
{   }
```

Примеры действий, после которых выполняются обработчики:

- Курсор мыши наведен на элемент;
- Веб-страница или картинка полностью загружена;
- Изменено содержимое поля формы;
- HTML-форма отправлена;
- Нажата клавиша на клавиатуре.

Обработчики событий jQuery

Общий вид определения обработчиков jQuery:

```
$(селектор).обработчик_события(function() {
  код_обработчика_события
});
```

…или

```
document.getElementById(…).addEventListener(“событие”, function()
{
  код_обработчика_события
});
```

Обработчики событий jQuery. Пример

```
$(document).ready(function() {
    $("#but1").click(function() {
        alert("Вы нажали один раз на первую кнопку!");
    });
    $("#but2").dblclick(function() {
        alert("Вы нажали два раза на вторую кнопку!");
    });
});
```

Обработчики событий jQuery. Пример

```
$(document).ready(function() {
    $("p").mouseover(function() {
        $("p").css("color", "green")
    });
    $("p").mouseout(function() {
        $("p").css("color", "black")
    });
});
```

Обработчики событий jQuery

blur()	change()	click()
dblclick()	focus()	focusin()
focusout()	hover()	keydown()
keyup()	load()	mousedown()
mouseenter()	mouseleave()	mousemove()
mouseout()	mouseover()	mouseup()
ready()	resize()	scroll()
select()	submit()	unload()

Объект event

```
$(селектор).событие(function(event) {
   // Затем в коде обработчика вы можете обращаться
   // к его свойствам и методам следующим образом:
    console.log(event.data);
    event.preventDefault();
});
```

Объект **event** хранит информации о произошедшем событии.
Его необходимо явно передать в **обработчик** события.

Объект event

Свойство	Описание
currentTarget	Содержит имя DOM элемента, в котором произошло событие.
data	Содержит дополнительные данные переданные обработчику события во время привязки его к элементу.
pageX	Содержит координаты указателя мыши по оси X во время вызова события.
pageY	Содержит координаты указателя мыши по оси Y во время вызова события.
result	Содержит последнее значение возвращенное вызванным ранее обработчиком события.
target	Содержит имя DOM элемента, который вызвал событие.
timeStamp	Содержит количество прошедших с 1 Января 1970 года миллисекунд до вызова данного события.
type	Содержит тип (название) произошедшего события.
which	Содержит код кнопки, которая была зажата во время вызова данного

Объект event

Метод	Описание
isDefaultPrevented()	Позволяет узнать, был ли вызван метод preventDefault() для данного элемента.
isImmediatePropagationStopped()	Позволяет узнать был ли вызван метод stopImmediatePropagation() для данного элемента.
isPropagationStopped()	Позволяет узнать был ли вызван метод stopPropagation() для данного элемента.
preventDefault()	Предотвращает выполнение стандартного действия элемента.
stopImmediatePropagation()	Запрещает вызов остальных обработчиков события привязанных к элементу.
stopPropagation()	Останавливает "всплытие" вызова события к родительским элементам.

jQuery Эффекты

jQuery Эффекты

С помощью jQuery методов
fadeOut(), **fadeIn()** и **fadeTo()**
можно скрывать и отображать элементы анимированно:

```
// Позволяет постепенно скрыть выбранный элемент
$("селектор").fadeOut(скорость, функция);
// Позволяет постепенно отобразить выбранный элемент
$("селектор").fadeIn(скорость, функция);
// Позволяет постепенно скрыть/отобразить элемент
// до указанного значения прозрачности
```

jQuery Эффекты

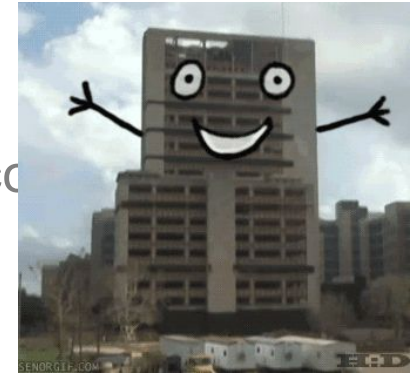
Пример:

```
$(document).ready(function() {
    $("#but1").click(function() { $("#par1").fadeOut(3000); });
    $("#but2").click(function() { $("#par1").fadeIn(3000); });
    $("#but3").click(function() { $("#par1").fadeTo(3000, 0.3); });
    $("#but4").click(function() { $("#par1").fadeTo(3000, 0.8); });
    $("#but5").click(function() { $("#par1").fadeOut(3000, function() {
        alert("Абзац был полностью скрыт.");
    });
});
```

jQuery Эффекты

С помощью методов **slideUp**, **slideDown** и **slideToggle** можно плавно изменять высоту выбранных элементов.

```
// Изменяет высоту элемента до нуля
$("селектор").slideUp(скорость, функция);
// Плавно возвращает элементу его изначальную высо
$("селектор").slideDown(скорость, функция);
// При первом вызове будет действовать как slideUp,
// а при втором как slideDown
$("селектор").slideToggle(скорость, функция);
```



jQuery Эффекты

Пример:

```
$(document).ready(function() {
    $("#but1").click(function(){ $("#square").slideUp(3000); });
    $("#but2").click(function(){ $("#square").slideDown(3000); });
    $("#but3").click(function(){ $("#square").slideToggle(3000); });
    $("#but4").click(function(){ $("#square").slideUp(3000, function() {
        alert("Текст был скрыт");
    });
});
```

Анимация в jQuery

```
$("селектор").animate(
    {стили}, скорость,
    функция_смягчения,

функция_обратного_вызова
);
```

- **стили** - CSS стили для анимации
- **скорость** - скорость анимации: "slow", "fast", "normal" или в мс
- **функция_смягчения** - функция, которая будет отвечать за плавность выполнения анимации
- **функция_обратного_вызова** - указывает имя функции, код которой будет выполнен после завершения анимации

Анимация в jQuery

Пример:

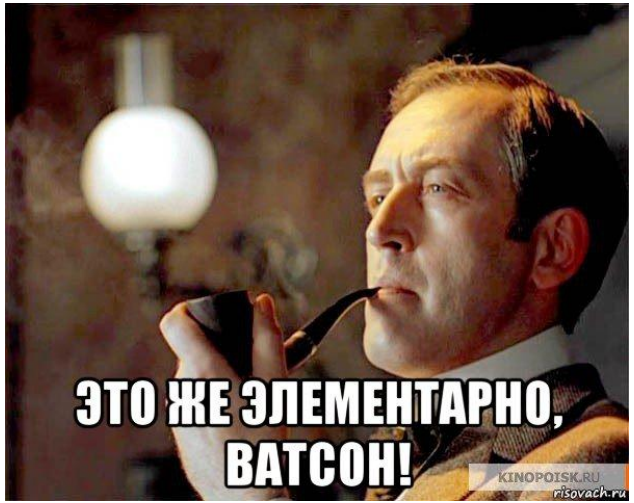
```
$(document).ready(function() {
  $("#but1").click(function() {
    $("#par1").animate({fontSize: "1.3em"}, 1000);
    $("#par1").animate({marginLeft: "30px"}, 1000);
    $("#par1").animate({marginTop: "50px"}, 1000);
    $("#par1").animate({fontSize: "1em"}, 1000);
    $("#par1").animate({marginLeft: "0px"}, 1000);
    $("#par1").animate({marginTop: "0px"}, 1000);
  });
});
```

Работа с DOM в jQuery

Изменение содержимого элементов с помощью jQuery:

```
// Узнаем содержимое элемента
var value = $("селектор").html();

// Изменим содержимое элемента
$("селектор").html("новое содержимое");
```



Работа с DOM в jQuery

append() / prepend() - вставить произвольный текст после или перед внутренним содержимым выбранного элемента

```
// Добавим текст после внутреннего содержимого элемента
$("селектор").append("произвольный текст");

// Добавим текст перед внутренним содержимым элемента
$("селектор").prepend("произвольный текст");
```

Работа с DOM в jQuery

Работа с атрибутами:
attr() - узнать/изменить содержимое атрибута у выбранного элемента.
removeAttr() - удалить указанный атрибут у выбранного элемента.

```
// Узнаем значение произвольного атрибута
var value = $("селектор").attr("атрибут");

// Установим новое значение произвольному атрибуту
$("селектор").attr("атрибут", "новое значение");

// Удалим атрибут
$("селектор").removeAttr("атрибут");
```

Работа с DOM в jQuery

```
$(document).ready(function(){
    $("#but1").click(function(){
        alert($("#anchor1").attr("href"));
    });
    $("#but2").click(function(){
        $("#anchor1").attr("href","http://www.kremlin.ru");
    });
    $("#but3").click(function(){
        $("#anchor1").removeAttr("href");
    });
});
```

Работа с DOM в jQuery

Метод **wrap** позволяет "обернуть" выбранный элемент указанными тэгами:

$("селектор").**wrap**("<нач_тэг><кон_тэг>");

Пример:

```
$(document).ready(function() {
	$("#but1").click(function(){
		$("#par1").wrap("<i></i>");
	});
	$("#but2").click(function(){
		$("#par3").wrap("<div id='wrap1'></div>");
	});
});
```

Управление стилями в jQuery

Управление стилями в jQuery

Узнать текущие или установить новые значения свойств стилей элементов:

```
// Узнаем значение указанного CSS свойства выбранного элемента
$("селектор").css("свойство");

// Установим новое значение указанному CSS свойству выбранного элемента
$("селектор").css("свойство", "значение");

// Установим произвольные значения нескольким CSS свойствам элемента
$("селектор").css({свойство1:значение1, свойствоN:значениеN});
```

Управление стилями в jQuery

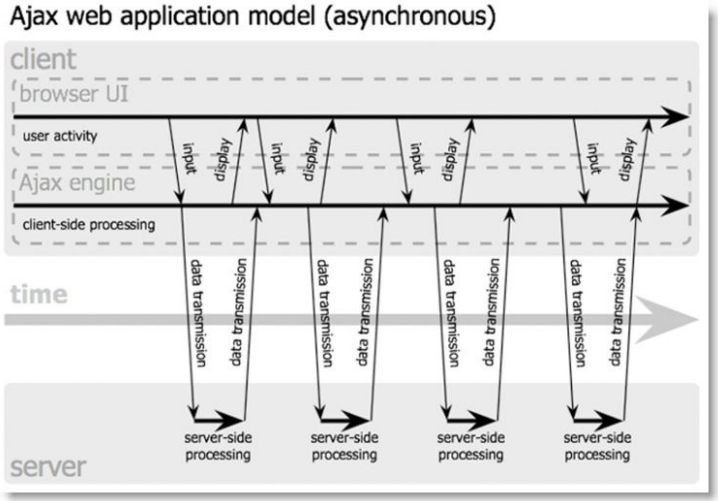
addClass() - добавить указанный класс выбранному элементу
removeClass() - удалить указанный класс у выбранного элемента
toggleClass() - переключение между удалением и добавлением класса

```
$("селектор").addClass("имя_класса");
$("селектор").removeClass('имя_класса');
$("селектор").toggleClass('имя_класса');
```

Асинхронная работа в jQuery

Использование асинхронных запросов позволяет значительно ускорить загрузку
страниц, т.к. обновляться будет только та часть страницы, которая содержит новые данные, а не страница целиком.

Техника использования асинхронных запросов называется **AJAX** - **A**synchronous **J**avaScript **A**nd **X**ML (Асинхронный JavaScript и XML)



Асинхронная работа в jQuery

Простейший способ асинхронного запроса:

$("селектор").**load**(url, данные, функция);

Пример:

```
$(document).ready(function() {
    $("#but1").click(function(){
        $("#par1").load("testfile.txt");
    })
});
```

Другие методы AJAX: http://www.wisdomweb.ru/JQ/ajax.php

За кадром

- Низкоуровневые методы работы с AJAX
- Плагины jQuery
- jQuery UI

Где почитать подробнее

- https://jquery.com/
- http://jquery.page2page.ru/
- http://www.wisdomweb.ru/JQ/jquery-first.php

Вопросы есть?

Спасибо за внимание!