# Intro

## Start

### Include library

<!--

http://jquery.com/ - официальный сайт библеотеки JQuery

http://api.jquery.com/ - документация

-->

<!DOCTYPE html>

<html>

<head>

<title>Знакомство с библиотекой JQuery. Подключение удаленной библиотеки.</title>

<!-- Подключение удаленной библиотеки (CDN) -->

<script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>

<!-- Подключение локальной библиотеки -->

<!-- <script src="jquery-1.10.2.js"></script> -->

<script>

jQuery(); // Главная функция для использования всех возможностей jQuery.

$(); // Псевдоним функции jQuery() с тем же набором функционала.

</script>

</head>

<body>

<p>JQuery - write less, do more.</p>

<p><a href="http://jquery.com/" target="\_blank">Official site</a></p>

<p><a href="http://api.jquery.com/" target="\_blank">Documenatation</a></p>

</body>

</html>

### Ready

<head>

<title>Метод ready().</title>

<script src="jquery-1.10.2.js"></script>

<script>

/\*

Событие ready происходит в момент готовности DOM-дерева, что происходит раньше начала загрузки "тяжелых" мультимедийных файлов.

Это оптимальный момент, когда можно приступить к установке обработчиков различных событий и выполнять другой подготовительный javascript-код. ВАЖНО!

Если подготовительные действия требуют наличия таких элементов как картинки,

правильным будет полагаться на событие load. Однако, имейте ввиду, что одновременное использование .ready() и <body onload="..."> приведет к ошибке.

Если вам необходимо использовать оба события, используйте метод библиотеки jQuery .load(), вместо <body onload="...">.

\*/

jQuery(document).ready(function () {

alert('Структура страницы сформирована, можно приступать!');

jQuery("p").text("Hello World (jQuery)");

});

// $() - псевдоним функции jQuery()

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

// alert('Структура страницы сформирована, можно приступать!');

// $("p").text("Hello World (jQuery)");

//});

</script>

</head>

<body>

<p></p>

</body>

</html>

## Selectors

### SelectAll

<title>Селектор \*</title>

<script src="jquery-1.10.2.js"></script>

<script>

/\*

$("\*") - выбор всех элементов на странице.

css("a","b") - метод jQuery, позволяющий добавлять CSS-стили елементам.

a - имя CSS-свойства.

b - значение.

\*/

$(document).ready(function () {

$("\*").css("font-style", "italic");

});

</script>

</head>

<body>

<p>Paragraph in body</p>

<div>

<p>Paragraph in div</p>

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

<li>Item 4</li>

</ul>

</div>

</body>

### Select Tag Name

<title>Селектор имя тега</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E") - соответствует всем элементам с именем тега E.

// Выбор всех параграфов на странице и установка СSS свойству color значения red.

$("p").css("color", "red");

});

</script>

</head>

<body>

<p>Paragraph in body</p>

<div>

<p>Paragraph in div</p>

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

<li>Item 4</li>

</ul>

</div>

</body>

### Multiple select

<title>Группировка</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E,G,H") - соответствует всем элементам с именами тегов E, G, H.

// Выбор тегов "p" и "ul" на странице.

$("p,ul").css("color", "red");

});

</script>

</head>

### Select inherit

<title>Выбор вложенных элементов (потомков)</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E F") - соответствует всем элементам с именем тега F, вложенным в элемент с именем тега E.

// Выбор всех параграфов, вложенным в тег "div".

$("div p").css("color", "red");

});

</script>

</head>

### Neighborhood select

<head>

<title>Выбор соседних элементов.</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E + F") - соответствует всем элементам с именем тега F, которым непосредственно предшествует элемент E на том же уровне вложенности.

// Выбор всех элементов "div", которым непосредственно предшествует элемент "p" на том же уровне вложенности.

$("p + div").css("color", "red");

});

</script>

</head>

<body>

### Child select

<title>Выбор прямых потомков.</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E > F") - соответствует всем элементам с именем тега F, являющимися прямыми потомками элементов с именем тега E.

// Выбор всех элементов "span", являющимися прямыми потомками элементов "div".

$("div > span").css("color", "red");

});

</script>

</head>

### Has selector

<title>"Has"</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E:has(F)) - соответствует всем элементам с именем тега E, имеющим хотя бы один вложенный элемент с именем тега F.

// Выбор всех элементов с именем тега div, имеющим хотя бы один вложенный элемент с именем тега ul.

$("div:has(ul)").css("background", "red");

});

### Class select

<title>Выборка по имени класса</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E.C) - соответствует всем элементам E с именем класса C. При отсутствии E эквивалентен селектору \*.C

// Выбор всех элементов "p" с именем класса "red".

$("p.red").css("color", "red");

});

### Id select

<title>Выборка по идентификатору</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E#I) - соответствует всем элементам E с именем идентификатора I.

// Выбор всех элементов "p" с идентификатором "red".

$("p#red").css("color", "red");

});

### Sibling select

<title>Выборка "братьев".</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E ~ F") - соответствует всем элементам с именем тега F, которым предшествует элемент E на том же уровне вложенности.

// Выбор всех элементов "p", которым предшествует элемент "span" на том же уровне вложенности.

$("span ~ p").css("color", "red");

});

</script>

### Select attribute1

<title>Выборка по атрибутам.</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E[A]") - соответствует элементам E с атрибутом A, имеющим любое значение.

// Выбор всех элементов "a", с атрибутом "target", имеющим любое значение.

$("a[target]").css("color", "red");

});

### Select attribute2

<title>Выборка по атрибутам.</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

// $("E[A=V]") - соответствует элементам E с атрибутом A, который равен V.

// Выбор всех элементов "a", с атрибутом "target", имеющим значение text.

// Метод val() - метод, который позволяет получить значение атрибута "value" первого элемента в наборе соответствующих селектору элементов или установить новое значение (переданное в качестве аргумента) для всех елементов, подходящих под выборку.

$("input[type=text]").val("admin");

});

### Select attribute3

<title>Выборка по атрибутам.</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(document).ready(function () {

/\*

$("E[A^=V]") - соответствует элементам E с атрибутом A, который начинается с V.

$("E[A$=V]") - соответствует элементам E с атрибутом A, который заканчивается на V.

$("E[A\*=V]") - соответствует элементам E с атрибутом A, который содержит V.

\*/

$("a[href^='http://']").css("color", "red"); // Выбор всех элементов "a", с атрибутом "href", который начинается c "http://"

$("a[href$='.com']").css("color", "blue"); // Выбор всех элементов "a", с атрибутом "href", который заканчивается на ".com"

$("a[href\*='html']").css("color", "green"); // Выбор всех элементов "a", с атрибутом "href", который содержит "html"

});

</script>

<style>

span {

display: inline-block;

width: 80px;

}

</style>

</head>

<body>

<a href="http://edu.cbsystematics.com/video/ru">Видео обучение</a><br />

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

<a href="http://htmlbook.ru/">HTML-book</a>

</body>

</html>

## Samples

### Eq method

<title>Выборка элементов с помощью jQuery</title>

<script src="jquery-1.10.2.js"></script>

<script>

// eq() - метод, который позволяет выбрать нужный элемент из массива по индексу.

jQuery(document).ready(function () {

$('p').eq(0).text("$('p') - выборка по имени тега");

$('[name = "input\_background"]').css('background', 'yellow');

$('.bold\_paragraph').css('fontWeight', 'bold');

$('#red\_paragraph').css('color', 'red');

});

</script>

</head>

<body>

<p></p>

<input type="text" name="input\_background" />

<p class="bold\_paragraph"><u>$('.bold\_paragraph')</u> - выборка по имени класса.</p>

<p id="red\_paragraph"><u>$('#red\_paragraph')</u> - выборка по идентификатору</p>

</body>

</html

### Query selector method

<title>Выборка элементов с помощью методов querySelector() и querySelectorAll()</title>

<script>

window.addEventListener("load", function () {

// querySelector() - получение первого элемента на странице, который соответствует селектору.

// В данном случае возвращаем элемент p.

var paragraph = document.querySelector("p");

paragraph.style.fontSize = "30px";

// querySelectorAll() - возвращает массив элементов, которые соответствуют селектору.

// В данном случае возвращается массив элементов списка, находящихся в элементе ul с атрибутом class="list"

var list = document.querySelectorAll(".list > li");

for (var i = 0; i < list.length; i++) {

list[i].style.color = "blue";

}

});

</script>

</head>

<body>

<p>Метод querySelector()</p>

<p>Параграф без стилей</p>

<ul class="list">

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

<li>Item 4</li>

</ul>

</body>

</html>

### On method

ad>

<title>Добавление классов к елементам с помощью jQuery</title>

<script src="jquery-1.10.2.js"></script>

<script>

// on() - Добавление обработчика событий текущему выбранному элементу. Он обеспечивает все функциональные возможности, необходимые для присоединения обработчиков событий.

// addClass() - Добавление указанного класса(ов) к каждому из набора соответствующих элементов.

// removeClass() - Удаляет классы в наборе соответствующих элементов.

$(document).ready(function () {

$('#first').on('click', firstButtonHandler);

$('#second').on('click', secondButtonHandler);

var div = $('#testDiv');

function firstButtonHandler() {

div.removeClass();

div.addClass("class1");

}

function secondButtonHandler() {

div.removeClass();

div.addClass("class2");

}

});

</script>

<style>

.class1 {

font-family: Tahoma;

color: Blue;

}

.class2 {

font-family: Courier New;

color: Red;

}

</style>

</head>

<body>

<input id="first" type="button" value="First" />

<input id="second" type="button" value="Second" />

<div id="testDiv">

Some text

</div>

</body>

# Filters

## Basic filters

### EvenOdd

<title>Выбор четных и нечетных элементов</title>

<script src="jquery-1.10.2.js"></script>

<script>

// $("E:even"); - выбор всех четных по значению индекса элементов.

// $("E:odd"); - выбор всех нечетных по значению элементов.

$(function () {

$("div p:even").css("background-color", "red");

//$("p:odd").css("background-color", "red");

});

</script>

<style>

body {

background-color: #44e3ff;

color: #fff;

}

p {

padding: 10px;

font-weight: bold;

}

</style>

</head>

<body>

<div>

<p>Параграф 0</p>

<p>Параграф 1</p>

<p>Параграф 2</p>

<p>Параграф 3</p>

<p>Параграф 4</p>

<p>Параграф 5</p>

<p>Параграф 6</p>

<p>Параграф 7</p>

<p>Параграф 8</p>

</div>

</body>

### FirstLast

<title>Фильтры :first и :last</title>

<script src="jquery-1.10.2.js"></script>

<script>

// $("E:first") - выбор первого совпадения на странице.

// $("E:last") - выбор последнего совпадения на странице.

$(function () {

$("li:first").css("color", "blue");

$("li:last").css("color", "red");

});

</script

### FirstChildLastChild

<title>Фильтры :first-child и :last-child</title>

<script src="jquery-1.10.2.js"></script>

<script>

// $("E:first-child") - выбор первого дочернего элемента.

// $("E:last-child") - выбор последнего дочернего элемента.

$(function () {

$("li:first-child").css("color", "blue");

$("li:last-child").css("color", "red");

});

</script>

</head>

<body>

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

<li>Item 4</li>

<li>Item 5</li>

</ul>

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

<li>Item 4</li>

<li>Item 5</li>

</ul>

</body>

### Only child

<title>Фильтр :only-child</title>

<script src="jquery-1.10.2.js"></script>

<script>

// $("E:only-child") - выбор единственных дочерних элементов.

$(function () {

$("div p:only-child").css("color", "blue");

});

</script>

<style>

div {

border: 1px solid;

margin-bottom: 10px;

padding: 10px;

}

</style>

</head>

<body>

<div>

<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.</p>

</div>

<div>

<p>Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.</p>

<p>Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudium lectorum. Mirum est notare quam littera gothica, quam nunc putamus parum claram, anteposuerit litterarum formas humanitatis per seacula quarta decima et quinta decima. Eodem modo typi, qui nunc nobis videntur parum clari, fiant sollemnes in futurum.</p>

</div>

</body>

### NthChild

<script>

/\*

Фильтр :nth-child используется для добавления стиля к элементам на основе нумерации в дереве элементов.

:nth-child(n) - Выбирает n-ый дочерний элемент

:nth-child(even) - Выбирает все четные дочерние элементы

:nth-child(odd) - Выбирает все нечетные дочерние элементы

:nth-child(an+b) - В круглых скобках выражение, где a и b целые числа, а n — счетчик, который автоматически принимает значение 0,1,2...

\*/

$(function () {

$(".one tr:nth-child(7)").css("background-color", "lightblue");

$(".two tr:nth-child(even)").css("background-color", "lightblue");

$(".three tr:nth-child(odd)").css("background-color", "lightblue");

$(".four tr:nth-child(3n+1)").css("background-color", "lightblue");

});

</script>

<style>

.one, .two, .three, .four {

border: 1px solid;

border-collapse: collapse;

width: 300px;

margin-bottom: 30px;

text-align: center;

}

td, th {

border: 1px solid;

}

</style>

</head>

### Select less then index or more

<title>Фильтры eq(), gt(), lt()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// :eq(индекс) - выбор элемента по индексу.

// :gt(индекс) - все элементы, индекс которых больше указаного.

// :lt(индекс) - все элементы, индекс которых меньше указаного.

$(function () {

$("p:eq(3)").css("background-color", "#4dea4d"); // (equal) выделит параграф с индексом 3

//$("p:gt(4)").css("background-color", "#4dea4d"); // (greater than) выделит все параграфы от элемента с индексом 4 (4 > ...).

//$("p:lt(4)").css("background-color", "#4dea4d"); // (less than) выделит все параграфы до элемента с индексом 4 (4 < ...).

});

</script>

</head

### Eq example

// eq(n) - выбирает элемент с индексом n из набора. Индекс начинается с 0.

var i = 0;

$(function () {

// установим интервал в одну секунду

var timer = setInterval(choose, 1000);

});

function choose() {

i++;

// выделит все элементы списка до элемента с индексом i

$("li:eq(" + (i - 1) + ")")

.css("background-color", "white");

// выделит элементы списка по индексу i

$("li:eq(" + i + ")")

.css("background-color", "yellow");

}

</script>

<style>

ol li:first-child {

background-color: yellow;

}

</style>

</head>

<body>

<ol>

<li>Some item</li>

<li>Some item</li>

<li>Some item</li>

<li>Some item</li>

<li>Some item</li>

</ol>

### Not(n)

<script>

// :not(n) - выберет все элементы из массива выборки кроме тех, которые соответствуют "n".

$(function () {

// выберет все элементы из массива выборки кроме элемента с индексом 3.

$("li:not(li:eq(3))").css("background-color", "yellow");

});

</script>

</head>

### Header

<script>

// :header - получение всех заголовков.

$(function () {

//$("h1, h2, h3, h4, h5, h6").css("background-color", "#4dea4d");

$(":header").css("background-color", "#4dea4d");

});

</script>

</head>

<body>

<h1>Header 1</h1>

<h2>Header 2</h2>

<h3>Header 3</h3>

<h4>Header 4</h4>

<h5>Header 5</h5>

<h6>Header 6</h6>

</body>

### Animated

<script>

// $(":animated") - возвращает элементы, в которых происходит анимация.

// метод hide(n) - скрывает выбранные элементы за n секунд.

$(function () {

$(".last").hide(3000);

$(":animated").css("background", "red");

});

</script>

<style>

div {

width: 200px;

height: 200px;

border: 1px solid;

float: left;

margin-right: 30px;

}

</style>

</head>

<body>

<div></div>

<div></div>

<div class="last"></div>

</body>

## Content Filters

### Contains

<title>Фильтр :contains('text')</title>

<script src="jquery-1.10.2.js"></script>

<script>

//:contains('text') - фильтр, который выбирает все элементы, в которых встречается указанный текст.

$(function () {

$("p:contains('JavaScript')").css("background-color", "red");

//$("p:contains('HTML')").css("background-color", "red");

});

</script>

</head>

<body>

<p>JavaScript — прототипно-ориентированный сценарный язык программирования. Является диалектом языка ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.</p>

<p>HTML (от англ. HyperText Markup Language — «язык разметки гипертекста»;) — стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц создаются при помощи языка HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме. HTML является приложением («частным случаем») SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879. XHTML же является приложением XML.</p>

</body>

### Empty

<script src="jquery-1.10.2.js"></script>

<script>

//:empty - выбирает все элементы в которых отсутствует содержимое

$(function () {

$("div:empty").css("background-color", "Red");

});

</script>

<style>

div {

width: 200px;

height: 100px;

background-color: azure;

border: 1px solid gray;

margin-bottom: 10px;

}

</style>

</head>

<body>

<div>Some content</div>

<div></div>

<div>Some content</div>

## Attribute filters

### Attribute name equals value

<title>Фильтр атрибутов</title>

<script src="jquery-1.10.2.js"></script>

<script>

// $("E[AttrName~='value']"); - выбор всех элементов E с атрибутом AttrName содержащим value в списке значений, разделенных пробелом

$(function () {

$("img[alt~='new']").css("border", "4px solid #005cf2");

});

</script>

</head>

<body>

<img src="images/cat.jpg" alt="new\_cat" />

<img src="images/cat.jpg" alt="new cat" />

<img src="images/cat.jpg" alt="new-cat" />

</body>

<script src="jquery-1.10.2.js"></script>

<script>

// $("E[AttrName!='value']"); - выбор всех элементов E с атрибутом AttrName не равного value

$(function () {

$("a[target=\_blank]").css("background-color", "#9bff70");

//$("a[target!=\_blank]").css("background-color", "#9bff70");

});

</script>

</head>

<body>

<a href="http://edu.cbsystematics.com/" target="\_blank">Live education</a><br />

<a href="http://edu.cbsystematics.com/video/ru">Video education</a>

</body>

</html>

## Form filters

### Button filter

<title>Фильтр :button</title>

<script src="jquery-1.10.2.js"></script>

<script>

// :button - фильтр, который выбирает все кнопки на странице.

$(function () {

$(":button").css("font-weight","bold");

});

</script>

</head>

<body>

<form>

Login <input type="text" value="" /><br />

Password <input type="text" value="" /><br />

<input type="submit" value="Send" /><br />

<input type="button" value="Show" />

</form>

</body>

### Checkbox filters

<script src="jquery-1.10.2.js"></script>

<script>

// :checkbox - фильтр для выбора всех элементов "checkbox".

$(function () {

var count = $(":checkbox").length;

$(".output").text("Количество элементов \"checkbox\" на странице - " + count);

});

</script>

</head>

### Checked

<script>

// Свойство length - количество элементов в массиве.

// :checked фильтр выбирает только отмеченные флажки или переключатели

$(function () {

var countChecked = function () {

var n = $("input:checked").length;

$(".output").text("Пунктов отмечено - " + n);

};

countChecked();

$("input[type=checkbox]").on("click", countChecked);

});

</script>

</head>

<body>

<form>

<p><strong>Выбор пунктов:</strong></p>

Пункт 1

<input type="checkbox" value="p1" /><br />

Пункт 2

<input type="checkbox" value="p2" /><br />

Пункт 3

<input type="checkbox" value="p3" /><br />

Пункт 4

<input type="checkbox" value="p4" /><br />

<p class="output"></p>

</form>

### Disabled

<script>

// :disabled фильтр выбирает только элементы форм, находящиеся в неактивном состоянии

$(function () {

$("input:disabled").val("disabled");

});

</script>

</head>

<body>

<form>

<input type="text" disabled="disabled" />

<input type="text" />

</form>

</body>

### Enabled

<title>Фильтр :enabled</title>

<script src="jquery-1.10.2.js"></script>

<script>

// :enabled - фильтр выбирает только элементы форм, находящиеся в активном состоянии

$(function () {

$("input:enabled").val("enabled");

});

</script>

</head>

<body>

<form>

<input type="text" disabled="disabled" />

<input type="text" />

</form>

</body>

### File filter

<title>Фильтр :file</title>

<script src="jquery-1.10.2.js"></script>

<script>

// :file - выбирает все элементы типа "file"

$(function () {

$(":file").css("border", "1px solid red");

});

</script>

<style>

input {

display: block;

}

</style>

</head>

<body>

<form>

Name <input type="text" />

Surname <input type="text" /><br />

<input type="file" />

<input type="submit" />

</form>

</body>

# Work with attributes, properties and styles

## Attributes

### Attr

<title>Получение значения атрибутов</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод attr() позволяет получить доступ к HTML атрибуту первого тэга на странице, который был выбран селектором.

// Если элемент не имеет такого атрибута, то возвращается значение undefined.

// Также есть возможность добавить новый атрибут с помощью даного метода.

$(function () {

// Получение значения атрибута style у первого элемента, который найден селектором "p"

var style = $("p").attr("style");

var p = $("#output");

p.text("<b>Значение атрибута \"style\"</b> - " + style);

});

</script>

</head>

<body>

<p style="color: blue; font-weight: bold;">

Test string

</p>

<div id="output">

</div>

</body>

### Atrr key value

<script src="jquery-1.10.2.js"></script>

<script>

// Добавление/изменение атрибута с помощью метода attr().

// attr("имя\_атрибута","значение");

$(function () {

// Установка значеня атрибута "class" у всех параграфов

$('input').on("click", changeColor);

});

function changeColor()

{

$("p").attr("class", "testClass");

}

</script>

<style>

.testClass {

color: Red;

}

</style>

</head>

<body>

<input type="button" value="Ok ">

<p>

Test string 1

</p>

<p class="testClass">

Test string 2

</p>

<p>

Test string 3

</p>

</body>

</html>

### Attr Json

<script src="jquery-1.10.2.js"></script>

<script>

// В методы JQuery можно передавать JSON-объекты. С ними намного проще работать.

$(function () {

// Передача в метод attr() JSON-объекта

$("p").attr({

style: "color:red; font-size:22pt",

title: "JQuery Attr Test"

});

});

</script>

</head>

<body>

<p>

Test string

</p>

<p>

Test string 2

</p>

</body>

### Attr function

<script src="jquery-1.10.2.js"></script>

<script>

// Установка значения атрибута style через вызов функции у первого тэга! <p>

$(function () {

$("p:first").attr("style", CalculateValue());

});

function CalculateValue() {

return "color:red";

}

</script>

</head>

<body>

<p>

Test string 1

</p>

<p>

Test string 2

</p>

<p>

Test string 3

</p>

</body>

### Atrr remove

<script src="jquery-1.10.2.js"></script>

<script>

// Метод removeAttr() - удаляет указаный атрибут элемента.

$(function () {

// Удаление атрибута style у всех нечетных элементов <p>

$("p:odd").removeAttr("style");

});

</script>

</head>

<body>

<p style="color: Blue;">

Test string 1

</p>

<p style="color: Blue;">

Test string 2

</p>

<p style="color: Blue;">

Test string 3

</p>

<p style="color: Blue;">

Test string 4

</p>

</body>

### Prop

<script src="jquery-1.10.2.js"></script>

<script>

// http://javascript.ru/tutorial/dom/attributes - рекомендовано к прочтению

// Метод prop() - возвращает или изменяет значение свойств выбранных элементов страницы.

// .prop(propName, value) - во всех выбранных элементах, свойство propName примет значение value.

// .prop({propName1:value1, propName2:value2, ...}) - во всех выбранных элементах изменит значения группы свойств propName1, propName2, ... сделав их равными value1, value2, ... соответственно.

// .prop(propName, function(index, value)) - свойству propName будет присвоено значение, возвращенное пользовательской функцией (если она ничего не вернет, то свойство просто не изменит своего значения). Функция вызывается отдельно, для каждого из выбранных элементов. При вызове ей передаются следующие параметры: index — позиция элемента в наборе, value — текущее значение свойства propName у элемента.

// Метод html() - получает HTML-содержимое первого элемента в наборе соответствующих элементов или устанавливает HTML-разметку для каждого совпавшего элемента.

$(function () {

$("input").change(function () {

var $input = $(this);

$("p").html(

".attr( \"checked\" ): <b>" + $input.attr("checked") + "</b><br>" +

".prop( \"checked\" ): <b>" + $input.prop("checked") + "</b><br>") + "</b>";

}).change();

});

</script>

<style>

p {

margin: 20px 0 0;

}

b {

color: cadetblue;

}

</style>

</head>

<body>

<label for="check1">Сохранить пароль? </label>

<input id="check1" type="checkbox" checked="checked" title="Сохранить/не сохранять пароль">

<p></p>

### RemoveProp

<title>Метод removeProp()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод removeProp() - удаляет свойства у выбранных элементах страницы.

// Замечание: используйте этот метод для удаления искусственно введенных свойств, не стоит удалять нативные свойства элементов (т.е. первоначальные, родные свойства элементов), это может вызвать ошибки работы браузера. В частности, не используйте .removeProp() для удаления таких атрибутов как checked и disabled. Когда эти атрибуты отсутствуют в input-элементах, их одноименные свойства не перестают существовать, они просто принимают значение false.

// Метод append() - добавляет контент внутрь каждого элемента набора. Добавляемый контент следует за уже существующим. Данный метод подобен применению appendChild.

$(function () {

var output = $("p");

output.prop("secretCode", 1234);

output.append("Секретный код: ", String(output.prop("secretCode")), ". ");

output.removeProp("secretCode");

output.append("Теперь: ", String(output.prop("secretCode")), ". ");

});

</script>

</head>

<body>

<p></p>

</body>

### Val

<title>Метод val()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод val() - позволяет получать и изменять значения элементов форм.

// .val() - возвращает значение атрибута value у выбранного элемента формы. Если выбрано несколько элементов, то значение будет взято у первого. В случае элемента формы <select multiple="multiple"> возвращается массив всех выбранных значений.

// .val(newVal) - атрибуту value будет присвоено значение newVal, у всех выбранных элементов.

// .val(function(index, newVal) - атрибуту value будет присвоено значение, возвращенное пользовательской функцией. Функция вызывается отдельно, для каждого из выбранных элементов. При вызове, ей передаются следующие параметры: index — позиция элемента в наборе, newVal — текущее значение атрибута value у элемента.

$(function () {

//Нажатие на первую кнопку

$("#button1").click(function () {

$("#edit1").val("Hello world"); // Запись в поле ввода.

});

//Нажатие на вторую кнопку

$("#button2").click(function () {

alert($("#edit1").val()); // Чтение из поля ввода.

});

});

</script>

</head>

<body>

<input type="text" id="edit1" />

<input type="button" id="button1" value="Push me!" />

<input type="button" id="button2" value="Get Text" />

</body>

### Add Class

<script src="jquery-1.10.2.js"></script>

<script>

// Метод addClass() - добавляет класс(ы) к выбранным элементам страницы и не заменяет уже существующие.

// .addClass("clName1 clName2 ...") - добавляет классы clName1, clName2, ... всем выбранным элементам.

// .addClass(function(index, class)) - добавляет классы, список которых возвращает заданная пользователем функция. Возвращаемое ей значение должно представлять из себя строку, с перечислением добавляемых классов через пробел. Функция вызывается отдельно для каждого из выбранных элементов. При вызове ей передаются следующие параметры: index — позиция элемента в наборе, class — текущий класс(ы) элемента.

$(function () {

// Добавление класса к элементам с идентификаторами First, Second и Third.

$("#First").addClass("testClassGreen");

$("#Second").addClass("testClassRed");

$("#Third").addClass("testClassBlue");

});

</script>

<style>

.testClassGreen {

color: Green;

}

.testClassRed {

color: Red;

}

.testClassBlue {

color: Blue;

}

</style>

</head>

<body>

<p id="First">

Test string 1

</p>

<p id="Second">

Test string 2

</p>

<p id="Third">

Test string 3

</p>

</body>

### Remove Class

<title>Метод removeClass()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод removeClass() - удаляет заданные классы у элементов на странице.

// .removeClass() - удаляет все классы у выбранных элементов.

// .removeClass("class1 class2 ...") - удаляет классы class1, class2, ... у выбранных элементов.

// .removeClass(function(index, class)) - удаляет классы, список которых возвращает заданная пользователем функция. Возвращаемое ей значение должно представлять из себя строку с перечислением классов через пробел. Функция вызывается отдельно для каждого из выбранных элементов. При вызове ей передаются следующие параметры: index — позиция элемента в наборе, class — текущий класс(ы) элемента.

$(function () {

$(".class1").removeClass("class1");

});

</script>

### Toggle class

<script>

// Метод toggleClass() - добавляет или удаляет заданный класс(ы) по принципу переключателя (добавляет, если элемент не содержит класса, и удаляет, если класс есть).

// .toggleClass("class1 class2 ...") - соответствует поведению описанному выше.

// .toggleClass("class1 class2 ...", switch ) - только добавляет или только удаляет классы в зависимости от параметра switch. Параметр имеет тип boolean. true — в случае добавления класса(ов), false — иначе.

// .toggleClass( function(index, class) ,[switch] ) - "переключает" классы, список которых возвращает заданная пользователем функция. Возвращаемое ей значение должно представлять из себя строку, с перечислением классов через пробел. Функция вызывается для каждого из выбранных элементов. При вызове ей передаются следующие параметры: index — позиция элемента в наборе, class — текущий класс(ы) элемента. Параметр switch играет такую же роль, как и в предыдущем случае.

$(function () {

$("#toggle").on("click", toggle);

function toggle() {

$("p").toggleClass("class1");

}

});

</script>

### Has Class

<script src="jquery-1.10.2.js"></script>

<script>

// hasClass(className) - проверяет наличие класса "className" у выбранных элементов страницы. Если хотя бы один из элементов содержит этот класс, то функция вернет true, иначе вернет false.

$(function () {

var className = "class1";

if ($("p:first").hasClass(className)) {

alert("Тэг содержит указанный CSS-класс - " + className);

}

else {

alert("Тэг не содержит указанный CSS-класс");

}

});

</script>

## Css

### Css()

<title>Метод css()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод css() - возвращает или изменяет значения css-величин у выбранных элементов страницы.

// .css(styleName) - возвращает значение css-свойства styleName у выбранного элемента. Если выбрано несколько элементов, то значение будет взято у первого.

// .css(styleName, value) - css-свойству styleName будет присвоено значение value, у всех выбранных элементов.

// .css({styleName1: value1, styleName2: value2, ...}) - группе css-свойств styleName1, styleName2, ... будут присвоены значения value1, value2, ..., у всех выбранных элементов.

// .css(styleName, function(index, value)) - css-свойству styleName будет присвоено значение, возвращенное пользовательской функцией. Функция вызывается отдельно, для каждого из выбранных элементов. При вызове, ей передаются следующие параметры: index — позиция элемента в наборе, value — текущее значение css-свойства styleName у элемента.

$(function () {

$("p").css("color", "green");

});

</script>

</head>

<body>

<p>

Test string 1

</p>

<p>

Test string 2

</p>

<p>

Test string 3

</p>

// .css({styleName1: value1, styleName2: value2, ...}) - группе css-свойств styleName1, styleName2, ... будут присвоены значения value1, value2, ..., у всех выбранных элементов.

// .css(styleName, function(index, value)) - css-свойству styleName будет присвоено значение, возвращенное пользовательской функцией. Функция вызывается отдельно, для каждого из выбранных элементов. При вызове, ей передаются следующие параметры: index — позиция элемента в наборе, value — текущее значение css-свойства styleName у элемента.

$(function () {

$("p").css(

{

color: "green",

backgroundColor: "lightgray"

});

});

</script>

### Height

<script src="jquery-1.10.2.js"></script>

<script>

/\* Методы .height() .innerHeight() .outerHeight([includeMargin]) - все три функции возвращают высоту первого из выбранных элементов страницы.

height() — высота элемента без учета внутренних отступов и толщины рамки.

innerHeight() — высота элемента с учетом размера внутренних отступов (padding).

outerHeight(includeMargin) — высота элемента с учетом внутренних отступов, рамки (border-width) и при необходимости внешних отступов (includeMargin == true) \*/

// .height(value) - устанавливает новое значение высоты равное value, для всех выбранных элементов.

// .height(function(index, value)) - устанавливает новое значение высоты элементов, равное значению, которое вернет пользовательская функция. Функция вызывается отдельно, для каждого из выбранных элементов. При вызове ей передаются следующие параметры: index — позиция элемента в наборе, value — текущее значение высоты элемента.

$(function () {

var a = "";

a += "heigh() - " + $("div").height() + "<br />";

a += "innerHeight() - " + $("div").innerHeight() + "<br />";

a += "outerHeight() - " + $("div").outerHeight() + "<br />";

a += "outerHeight(true) - " + $("div").outerHeight(true) + "<br />";

$("p").html(a);

//$("div").height(400);

});

</script>

<style>

div {

width: 300px;

background: green;

padding: 10px;

margin: 15px;

color: white;

border: 5px solid yellow;

text-align: justify;

}

</style>

### Width

<title>Методы .width() .innerWidth() .outerWidth([includeMargin])</title>

<script src="jquery-1.10.2.js"></script>

<script>

/\* Методы .width() .innerWidth() .outerWidth([includeMargin]) - все три функции возвращают ширину первого из выбранных элементов страницы. Отличия у функций следующие:

width() — ширина элемента без учета отступов и толщины рамки.

innerWidth() — ширина элемента с учетом размера внутренних отступов (padding).

outerWidth(includeMargin) — ширина элемента с учетом внутренних отступов, рамки (border-width) и при необходимости внешних отступов (includeMargin == true) \*/

// .width(value) - устанавливает новое значение ширины равное value, для всех выбранных элементов.

// .width(function(index, value)) - устанавливает новое значение ширины элементов, равное значению, которое вернет пользовательская функция. Функция вызывается отдельно, для каждого из выбранных элементов. При вызове ей передаются следующие параметры: index — позиция элемента в наборе, value — текущее значение ширины элемента.

$(function () {

var a = "";

a += "width() - " + $("div").width() + "<br />";

a += "innerWidth() - " + $("div").innerWidth() + "<br />";

a += "outerWidth() - " + $("div").outerWidth() + "<br />";

a += "outerWidth(true) - " + $("div").outerWidth(true) + "<br />";

$("p").html(a);

//$("div").width(400);

});

</script>

<style>

div {

width: 300px;

background: green;

padding: 10px;

margin: 15px;

color: white;

border: 5px solid yellow;

text-align: justify;

}

</style>

### Offset

<title>Метод offset()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод offset() - возвращает координаты выбранного элемента относительно начала страницы. Также с помощью offset(), можно изменить координаты элемента.

// offset(value) - изменяет координаты всех выбранных элементов делая их равными value. Значение value должено быть объектом с двумя свойствами — {top:newTop, left:newLeft}.

// offset(function(index, value)) - устанавливает новое значение координат элементов, которое возвращает пользовательская функция. Функция вызывается отдельно для каждого из выбранных элементов. При вызове ей передаются следующие параметры: index — позиция элемента в наборе, value — текущие координаты элемента.

// Замечание 1: при изменении координат с помощью функции offset, все выбранные элементы, у которых не задан тип позиционирования (то есть position = static), автоматически изменят позиционирование на относительное (relative).

// Замечание 2: важно отметить, что используя метод .offset() вы получите позицию только первого элемента из всех выбранных.

$(function () {

var element = $("div").offset();

var output = $("div");

output.html("Мои координаты относительно окна браузера:<br />" + "x: " + element.left + "<br />y: " + element.top);

});

</script>

<style>

\* {

margin: 0;

padding: 0;

font-family: Arial, 'DejaVu Sans', 'Liberation Sans', Freesans, sans-serif;

font-size: 12px;

}

div {

background-color: #7b91e5;

width: 270px;

height: 200px;

position: absolute;

top: 50px;

left: 200px;

padding: 5px;

}

</style>

</head>

<body>

### Position

<script src="jquery-1.10.2.js"></script>

<script>

// Метод position() - возвращают координаты выбранного элемента относительно ближайшего родителя, у которого задан тип позиционирования (css-свойство position равно relative или absolute или fixed).

$(function () {

var element = $("#child").position();

var output = $("#output");

output.html("Мои координаты относительно родителя:<br />" + "x: " + element.left + "<br />y: " + element.top);

});

</script>

<style>

\* {

margin: 0;

padding: 0;

font-family: Arial, 'DejaVu Sans', 'Liberation Sans', Freesans, sans-serif;

font-size: 12px;

}

/\*#main\_parent{

background-color: #f63737;

width: 700px;

height: 500px;

position: relative;

left: 300px;

top: 50px;

padding: 5px;

}\*/

#parent {

background-color: #c7cad6;

width: 500px;

height: 400px;

left: 500px; /\* Заменить на margin-left: 50px \*/

top: 100px; /\* Заменить на margin-top: 10px \*/

position: relative; /\* Закомментировать (Ctrl + k + c) \*/

padding: 5px;

}

#child {

background-color: #7b91e5;

width: 270px;

height: 200px;

position: absolute;

top: 50px;

left: 200px;

padding: 5px;

}

</style>

</head>

<body>

<!--<div id="main\_parent">-->

<div id="parent">

родитель

<div id="child">

<p id="output"></p>

</div>

</div>

<!--</div>-->

</body>

### Offset parent

<script src="jquery-1.10.2.js"></script>

<script>

// Метод offsetParent() - возвращает ближайшего предка (т.е. из числа: родитель, прародитель, и так далее, до начало дерева DOM), у которого задан тип позиционирования (position равен relative или absolute или fixed).

// Элементы с заданным позиционированием играют особую роль, поскольку именно относительно них происходит позиционирование всех элементов внутри.

$(function () {

$(".inner").offsetParent().css('background-color', 'yellow');

});

</script>

<style>

div {

border: 1px solid black;

}

.inner {

width: 30px;

height: 30px;

font-size: 20pt;

padding: 7px;

margin: 7px;

background-color: white;

}

.outer {

position: relative;

width: 100px;

height: 100px;

margin: 15px;

background-color: white;

}

.wrap {

width: 60px;

height: 60px;

margin: 18px;

background-color: white;

}

</style>

</head>

<body>

<div class="outer">

<div class="wrap">

<div class="inner">

B

</div>

</div>

</div>

</body>

# DOM

## Filters

### Eq()

<script src="jquery-1.10.2.js"></script>

<script>

// Метод eq(n) - возвращает элемент, идущий под индексом 'n' в наборе выбранных элементов.

$(function () {

$("div").eq(2).addClass("blue");

});

</script>

<style>

div {

width: 60px;

height: 60px;

margin: 10px;

float: left;

border: 2px solid blue;

}

.blue {

background: blue;

}

</style>

</head>

<body>

<div></div>

<div></div>

<div></div>

<div></div>

<div></div>

<div></div>

### Not()

title>Метод not()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод not(n) исключает из набора те элементы, которые соответствуют селектору 'n'.

$(function () {

$('li').not(':eq(2)').css("color","red");

});

</script>

</head>

<body>

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

<li>Item 4</li>

</ul>

</body>

</html>

### Is()

title>Метод is()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// is(n) - проверяет текущий набор элементов на соответствие указанному выражению 'n'. Возвращает true, если хотя бы один элемент соответствует выражению.

$(function () {

$("div").on("click", function () {

if ($(this).is(":first-child")) {

$("p").text("Это первый div.");

} else if ($(this).is(".blue,.red")) {

$("p").text("Это красный или синий div.");

} else if ($(this).is(":contains('Test')")) {

$("p").text("Содержит текст TEST!");

} else {

$("p").html("тут нет ничего особенного.");

}

});

});

</script>

<style>

div {

width: 60px;

height: 60px;

margin: 5px;

float: left;

border: 4px outset;

background: green;

text-align: center;

font-weight: bolder;

cursor: pointer;

}

.blue {

background: blue;

}

.red {

background: red;

}

span {

color: white;

font-size: 16px;

}

p {

color: red;

font-weight: bolder;

background: yellow;

margin: 3px;

clear: left;

}

</style>

</head>

<body>

<div></div>

<div class="blue"></div>

<div></div>

<div class="red"></div>

<div>

<br />

<span>Test</span>

</div>

<div class="blue"></div>

<p>&nbsp;</p>

### Filter

<script src="jquery-1.10.2.js"></script>

<script>

// Метод filter(selector) - в качестве параметра может быть передан селектор, используемый для получения соответствующих ему элементов.

$(function () {

var paragraphs = $("p");

paragraphs.filter(".test").css("background-color", "red");

paragraphs.filter(":odd").css("background-color", "green");

});

</script>

<style>

body {

background-color: Silver;

color: White;

}

p {

padding: 10px;

font-weight: bold;

background-color: Gray;

}

</style>

</head>

### Filter function

<title>Метод filter(function)</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод filter(function). Если передать методу filter() некоторую функцию, он вызовет ее для каждого элемента в наборе и исключит те элементы, для которых эта функция вернет значение false.

// Эта функция получает доступ к текущему элементу набора через $(this).

$(function () {

// В качестве параметра "index" функция будет получать порядковый номер обрабатываемого элемента.

// если функция возвращает значение true - элемент проходит проверку и попадает в результирующий набор данных

// $(this) - обращение к текущему элементу а функции

$("p").filter(function (index) {

if ($(this).attr("align") && index == 0) {

return true;

}

}).css("background-color", "red");

});

</script>

<style>

### Slice

<title>Метод slice(start,end) </title>

<script src="jquery-1.10.2.js"></script>

<script>

// slice(start,end) - выделяет подмножество из набора совпавших элементов.

// Ведет себя точно так же как и встроенный метод разделения массивов.

// Выделение подмножества начинается с индекса "start" (включительно) по индекс "end" (не включительно).

$(function () {

$("#btn1").click(function () {

$("p").css("background-color", "green");

$("p")

.slice(0, 3)

.css("background-color", "red");

});

$("#btn2").click(function () {

$("p").css("background-color", "blue");

$("p")

.slice(2, 4)

.css("background-color", "red");

});

$("#btn3").click(function () {

$("p").css("background-color", "green");

$("p")

.slice(3)

.css("background-color", "red");

});

$("#btn4").click(function () {

$("p").css("background-color", "green");

$("p")

.slice(3, -3)

.css("background-color", "red");

});

});

</script>

## Traversing

### Add

<title>Метод add(selector)</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод add(selector) - добавляет в уже существующий набор дополнительные элементы, которые соответсвуют указанному выражению.

$(function () {

$("p") // получаем все параграфы

.add("span") // добавляем к выборке span

.css("background-color", "Red");

});

</script>

<style>

body {

background-color: Silver;

color: White;

}

p {

padding: 10px;

font-weight: bold;

background-color: Gray;

}

</style>

</head>

<body>

<p>Test paragraph 1</p>

<p>Test paragraph 2</p>

<p>Test paragraph 3</p>

<p>Test paragraph 4</p>

<span>First Span</span>

<span>Second Span</span>

### Find

<script src="jquery-1.10.2.js"></script>

<script>

// Метод find(selector) - отыскивает дочерние элементы, соответствующие указанному выражению.

$(function () {

// находим все параграфы и ищем в них элементы span

$("p").find("span").css('color', 'red');

});

</script>

</head>

<body>

<p>Тестовый параграф с элементами <span>span</span> и <span>span</span></p>

<span>Просто span</span>

### Contens

<script src="jquery-1.10.2.js"></script>

<script>

// Метод contents() - получает потомков каждого элемента в наборе соответствующих элементов, включая текстовые узлы и узлы комментариев.

$(function () {

//Превратим текстовые узлы внутри #content в параграфы (текстовые узлы: nodeType = 3)

$("#content").contents().filter(function () {

return this.nodeType === 3;

}).wrap("<p></p>");

//Покрасим выше созданные параграфы внутри #content в красный цвет.

$("#content p").css("color", "red");

});

</script>

</head>

<body>

<div id="content">

lorem ipsum

<span>lorem ipsum in span</span>

</div>

### Children

<title>Метод children()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод children(selector) - получает набор элементов, содержащий всех прямых потомков для каждого совпавшего элемента.

$(function () {

$(".class1")

.children("p") // получить все дочерние элементы p для элемента с классом class1

.css("background-color", "Red");

});

</script>

</head>

### Hover

<script src="jquery-1.10.2.js"></script>

<script>

// parent() - возвращает ссылку на ближайшего родителя.

// hover(over,out) - метод, который имитирует эффект hovering (отслеживание попадания указателя мыши в пределы объекта). Параметры метода: over - функция при наведении, out - функция при отведении курсора мыши.

$(function () {

$("span").hover(

// функция срабатывает при наведении курсора на элемент

function () {

$(this).css("background-color", "Yellow")

.parent()

.css("background-color", "Red");

},

// функция срабатывает при выходе курсора за пределы элемента

function () {

$(this).css("background-color", "White")

.parent()

.css("background-color", "White");

});

});

</script>

</head>

<body>

<div>

<p>

Для изменения цвета этого параграфа нужно навести курсор мыши на этот <span>Span</span>

</p>

### Parents

<script src="jquery-1.10.2.js"></script>

<script>

/\* parents() - выбирает все родительские элементы (узлы) для ранее выбраных элементов. Принимает в качестве аргумента

дополнительный фильтр-селектор.\*/

$(function () {

$("span")

.parents("div")

.css("background-color", "Yellow");

//$("span")

// .parents()

// .css("background-color", "Yellow");

});

</script>

### Parents()

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

/\* parents() - выбирает все родительские элементы (узлы) для ранее выбраных элементов. Принимает в качестве аргумента дополнительный фильтр-селектор.\*/

function showParents() {

$("div").css("border-color", "white");

var len = $("span.selected")

.parents("div")

.css("border", "2px red solid")

.length;

$("b").text("Уникальных родителей (элементов div): " + len);

}

function customHover() {

$(this).toggleClass("selected");

showParents();

}

$("span").hover(customHover);

});

</script>

### Closet

script src="jquery-1.10.2.js"></script>

<script>

// Метод closest() - получает набор, содержащий ближайшие родительский элемент, которые соответствуют указанному селектору.

$(function () {

$("span").on("click", function () {

$(this).closest("div").toggleClass("hilight");

});

});

</script>

<style>

.outer {

padding: 20px;

background: red;

}

.inner {

padding: 20px;

background: yellow;

}

p {

padding: 5px;

background: green;

}

span {

background: pink;

}

.hilight {

background: blue;

}

</style>

</head>

<body>

<div class="outer">

<div class="inner">

<p>

<span>Show closest div</span>

</p>

</div>

</div>

### Next

<script src="jquery-1.10.2.js"></script>

<script>

// Метод next() - получает набор элементов, содержащий уникальные последующие родственные элементы для всех элементов существующего набора.

$(function () {

$("p:first").next().css("background", "yellow");

//$("p").next().css("background", "yellow");

});

</script>

</head>

<body>

<p>Paragraph 1</p>

<span>Span 1</span>

<p>Paragraph 2</p>

<span>Span 2</span>

<p>Paragraph 3</p>

<span>Span 3</span>

### NextAll

<script src="jquery-1.10.2.js"></script>

<script>

// Метод nextAll() - поиск всех родственных элементов после текущего элемента.

$(function () {

$("p.class1").nextAll().css("background-color", "Red");

//$("p.class1").nextAll("p").css("background-color", "Red");

});

</script>

<style>

body {

background-color: Silver;

color: White;

}

p {

padding: 10px;

font-weight: bold;

background-color: Gray;

}

</style>

</head>

<body>

<p>Paragraph 1</p>

<p>Paragraph 2</p>

<p class="class1">Paragraph 3</p>

<p>Paragraph 4</p>

<p>Paragraph 5</p>

<p>Paragraph 6</p>

<p>Paragraph 7</p>

<p>Paragraph 8</p>

<p>Paragraph 9</p>

<p>Paragraph 10</p>

<div>Div</div>

### NextUntil

<script src="jquery-1.10.2.js"></script>

<script>

// Метод nextUntil(selector) - Поиск всех родственных элементов после текущего элемента до указанного селектора.

$(function () {

//$(".class1").nextUntil().css("background-color", "Red");

$(".class1").nextUntil(".class2").css("background-color", "Red");

});

</script>

### Prev

<script src="jquery-1.10.2.js"></script>

<script>

// Метод prev() - получает набор элементов, содержащий уникальные предыдущие родственные элементы для всех элементов существующего набора.

$(function () {

$("p").prev().css("background", "yellow");

//$("p:last").prev().css("background", "yellow");

});

</script>

</head>

<body>

<span>Hello Again (span)</span>

<p>Hello (paragraph)</p>

<span>Hello Again (span)</span>

<p>And Again</p>

<span>Hello Again (span)</span>

<p>Hello (paragraph)</p>

### PrevAll

<script src="jquery-1.10.2.js"></script>

<script>

// Метод prevAll() - поиск всех родственных элементов перед текущим элементом

$(function () {

$("p.class1").prevAll().css("background-color", "Red");

//$("p.class1").prevAll("p").css("background-color", "Red");

});

</script>

### PrevUntil

<script src="jquery-1.10.2.js"></script>

<script>

// Метод prevUntil(selector) - поиск всех родственных элементов перед текущим элементом до указанного селектора.

$(function () {

//$(".class1").prevUntil().css("background-color", "Red");

$(".class2").prevUntil(".class1").css("background-color", "Red");

});

</script>

### Siblings

<script src="jquery-1.10.2.js"></script>

<script>

// Метод siblings() - получение всех родственных элементов.

$(function () {

var len = $(".highlight")

.siblings()

.css("color", "red")

.length;

$("b").text(len);

});

</script>

<style>

ul {

float: left;

margin: 5px;

font-size: 16px;

font-weight: bold;

}

p {

color: blue;

margin: 10px 20px;

font-size: 16px;

padding: 5px;

font-weight: bolder;

clear: both;

}

.highlight {

background: yellow;

}

</style>

</head>

<body>

<ul>

<li>One</li>

<li>Two</li>

<li class="highlight">Three</li>

<li>Four</li>

</ul>

### AdnSelf

<script src="jquery-1.10.2.js"></script>

<script>

// Метод .andSelf() добавляет предыдущий набор jQuery к текущему в пределах одной цепочки команд и возвращает объединенный набор.

$(function () {

$("div") // берем div'ы

.find("p") // находим дочерние параграфы

.andSelf() // добалвяем в выборку div'ы

.addClass("border"); // добавляем класс

//$("div").find("p").addClass("border");

});

</script>

### End

<script src="jquery-1.10.2.js"></script>

<script>

// end() - отменяет последнее деструктивное действие, возвращая тем самым набор элементов к его предыдущему состоянию (до деструктивной операции). Если никакой деструктивной операции не проводилось, то возвращается пустой набор. Под деструктивной операцией понимают операцию, которая изменяет набор элементов jQuery. То есть это любая из Traversing-функций, возвращающая объект jQuery: add, andSelf, children, filter, find, map, next, nextAll, not, parent, parents, prev, prevAll, siblings, и slice, а также clone, appendTo, prependTo, insertBefore, insertAfter, и replaceAll.

$(function () {

$("p") // находим все p

.add("div").css("border", "1px solid red") // добавляем к выборке все div и задаем рамку

.end().css("background-color", "green"); // убираем div, оставляя p и меняем фон

});

</script>

</head>

<body>

### Find example

script src="jquery-1.10.2.js"></script>

<script>

// find(selector) - отыскивает дочерние элементы, которые удовлетворяют указанному выражению.

$(function () {

// <span> When</span> <span>the</span> <span>day</span> <span>is</span> <span>short</span>

var newText = $("p") // находим все параграфы

.text() // извлекаем текст из параграфов

.split(" ") // разбиваем текст по пробелам

.join("</span> <span>"); // оборачиваем каждое слово в span

newText = "<span>" + newText + "</span>";

$("p").html(newText)

.find("span")

// на <span> устанавливаем обработчики на событие наведения/отведения курсора мыши

.hover(function () {

$(this).addClass("highlight");

},

function () {

$(this).removeClass("highlight");

})

// отменяем деструктивное действие метода find() и находим элементы, содержащие символ t. Добавляем им стили.

.end()

.find(":contains('t')")

.css(

{

"font-style": "italic",

"text-decoration": "underline"

});

});

</script>

<style>

p {

font-size: 20px;

width: 200px;

cursor: default;

color: blue;

font-weight: bold;

margin: 0 10px;

}

.highlight {

background: yellow;

}

</style>

</head>

<body>

<p>When the day is short find that which matters to you or stop believing</p>

</body>

</html>

## Insertion

### Html

<script src="jquery-1.10.2.js"></script>

<script>

//Метод html() - получает HTML-содержимое первого элемента в наборе соответствующих элементов или устанавливает HTML-разметку для каждого совпавшего элемента.

$(function () {

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

// $(this) - ссылка на объект, в котором произошло событие.

alert($(this).html());

});

});

</script>

<style>

p {

background-color: #bdb1c9;

color: White;

padding: 4px;

font-family: Verdana;

cursor: pointer;

}

</style>

</head>

<body>

<p>

Test string 1

</p>

<p>

<b>Test string 2</b>

</p>

<p>

Test string 3

</p>

</body>

### Html(val)

<title>Метод html(val)</title>

<script src="jquery-1.10.2.js"></script>

<script>

//Метод html() - получает HTML-содержимое первого элемента в наборе соответствующих элементов или устанавливает HTML-разметку для каждого совпавшего элемента.

$(function () {

$("p").html("<u>Hello world</u>");

});

</script>

<style>

p {

background-color: #bdb1c9;

color: White;

padding: 4px;

font-family: Verdana;

cursor: pointer;

}

</style>

</head>

<body>

<p>

Test string 1

</p>

<p>

<b>Test string 2</b>

</p>

<p>

Test string 3

</p>

### Text

<script src="jquery-1.10.2.js"></script>

<script>

//Метод text() - получает содержимое текста каждого элемента в наборе соответствующих элементов, в том числе их потомков, или устанавливает текстовое содержимое элементов набора.

$(function () {

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

alert($(this).text());

//alert($(this).html());

});

});

</script>

### Text(val)

<title>text()</title>

<script src="jquery-1.10.2.js"></script>

<script>

//Метод text() - получает содержимое текста каждого элемента в наборе соответствующих элементов, в том числе их потомков, или устанавливает текстовое содержимое элементов набора.

$(function () {

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

$(this).text("<a href=\"#\">Link</a>");

//$(this).html("<a href=\"#\">Link</a>");

});

});

</script>

### Append

<title>Методы append() и appendTo()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод append() - добавляет контент внутрь каждого элемента набора. Добавляемый контент следует за уже существующим. Данный метод подобен применению appendChild.

// Метод appendTo() - добавляет все элементы набора к другому указанному набору элементов. Добавляемые элементы следуют после уже существующих.

$(function () {

$("p").append("new text"); // Добавление контента в элемент "p".

$("table").append("<tr><td>3</td><td>4</td></tr>");

$("<b>Hello world</b>").appendTo(".test"); // Добавление контента в элемент ".test".

});

</script>

### Prepend

<title>Методы prepend() и prependTo()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод prepend() - добавляет контент внутрь каждого элемента набора. Добавляемый контент следует перед уже существующим.

// Метод prependTo() - добавляет все элементы набора к другому указанному набору элементов. Добавляемые элементы следуют перед уже существующими.

$(function () {

$("p").prepend("new text"); // Добавление контента в элемент "p".

$("table").prepend("<tr><td>3</td><td>4</td></tr>");

$("<b>Hello world</b>").prependTo(".test"); // Добавление контента в элемент ".test".

});

</script>

### Wrap/Empty/Remove

<title>Методы wrap(), empty(), remove().</title>

<script src="jquery-1.10.2.js"></script>

<script>

// wrap() - вставляет каждый совпавший элемент набора в указанную конструкцию HTML кода. Этот процесс наиболее полезен для встраивания дополнительной структуры в документ без необходимости разрушения его первоначальных семантических свойств.

// empty() - удаляет все содержимое из каждого элемента в наборе совпавших элементов.

// remove() - удаляет все совпавшие элементы из DOM. Эта функция НЕ удаляет элементы из объекта jQuery, позволяя Вам воспользоваться ими позже. Обратите внимание, начиная с версии 1.2.2 эта функция также удаляет все обработчики событий и внутренние закешированные данные.

$(function () {

$("span").wrap("<div>");

$("#clearBtn").click(function () {

$("p").empty();

});

$("#removeBtn").click(function () {

$("span").remove();

});

});

</script>

<style>

div {

background-color: Blue;

padding: 10px;

}

span {

background-color: White;

}

</style>

</head>

<body>

<span>Paragraph</span>

<p>Hello world</p>

<input type="button" value="Clear" id="clearBtn" />

<input type="button" value="Remove" id="removeBtn" />

</body>

</html>

# Events and Event handlers

## Events

### Ready event

<script>

// ready - определяет функцию, которая будет запущена после полной загрузки DOM

// Первый вариант.

$(document).ready(function () {

alert("Page Loaded I");

});

// Второй вариант.

$(document).ready(after\_ready);

function after\_ready() {

alert("Page Loaded II");

}

// Третий вариант вариант. (Сокращенная запись)

$(function () {

alert("Page Loaded III");

});

</script>

### Click

<title>Событие click</title>

<script src="jquery-1.10.2.js"></script>

<script>

// click() - добавляет обработчик на JavaScript событие "click"

// click(function) - добавляет функцию к событию click для каждого элемента набора.

// click() рассматривается как mousedown и mouseup, следующие друг за другом на одном и том же элементе.

$(function () {

$("p").click(function () { alert("clicked"); });

$("#button").click(function () { $("p").click() });

});

</script>

</head>

<body>

<p>Параграф №1</p>

<p>Параграф №2</p>

<p>Параграф №3</p>

<input type="button" id="button" value="push me" />

### Focus/blur

<title>События focus & blur</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function() {

// focus - событие происходит, когда элемент получает фокус

$("input").focus(function() {

$(this).val("focus");

});

// blur - событие происходит, когда элемент теряет фокус

$("input").blur(function() {

$(this).val("blur");

});

});

</script>

</head>

<body>

<input type="text" />

<br />

<br />

### KeyDown

<title>Событие keydown</title>

<script src="jquery-1.10.2.js"></script>

<script>

// keydown() - устанавливает обработчик события keydown для каждого элемента набора.

// event - объект, определяющий информацию о событии.

$(window).keydown(function (event) {

switch (event.keyCode) { // event.keyCode - код нажатой клавиши

case 37: {

alert("left pushed");

}

break;

case 38: {

alert("up pushed");

}

break;

case 39: {

alert("right pushed");

}

break;

case 40: {

alert("down pushed");

}

break;

}

});

</script>

### KeyUp

<title>Событие keyup</title>

<script src="jquery-1.10.2.js"></script>

<script>

// keyup() - создает обработчик на событие keyup для каждого элемента набора.

$(window).keyup(function (event) {

switch (event.keyCode) {

case 37: {

alert("left poped");

}

break;

case 38: {

alert("up poped");

}

break;

case 39: {

alert("right poped");

}

break;

case 40: {

alert("down poped");

}

break;

}

});

</script>

### KeyPress

<title>Событие keypress</title>

<script src="jquery-1.10.2.js"></script>

<script>

// keypress - событие, которое возникает при нажатии символьной клавиши клавиатуры (не срабатывает для функциональных клавиш (Shift, Ctrl, Alt и т.д.)

$(window).keypress(function (event) {

alert(String.fromCharCode(event.charCode) + "(" + event.charCode + ")");

});

</script>

### MouseOver/MouseOut

<title>События mouseover & mouseout</title>

<script src="jquery-1.10.2.js"></script>

<script>

// mouseover(f) - функция, которая принимает и назначает функцию "f" как обработчик на событие mouseover для каждого элемента набора.

// событие происходит при входе курсора в область элемента.

// mouseout(f) - функция, которая принимает и назначает функцию "f" как обработчик на событие mouseout для каждого элемента набора.

// событие происходит при выходе курсора в область элемента.

// Создаем счетчики для подсчета количества входов и выходов мыши за пределы элемента.

var overCount = 0;

var outCount = 0;

$(function () {

// Добавляем обработчики на события

$("div:first").mouseover(function () { overCount++; Update(); }) // курсор попадает в область элемента

$("div:last").mouseout(function () { outCount++; Update(); }) // курсор выходит за область элемента

});

function Update() {

$("#over").text("MouseOver " + overCount);

$("#out").text("MouseOut " + outCount);

}

</script>

</head>

<body>

<div style="padding: 20px; background-color: yellow; margin-bottom: 30px;">

<span id="over">MouseOver</span>

</div>

<div style="padding: 20px; background-color: lightsteelblue">

<span id="out">MouseOut</span>

</div>

</body>

### Hover

<title>Функция hover()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// hover(in, out) - функция, которая отвечает за привязку обработчиков на события mouseover и mouseout.

$(function () {

$("div").hover(onIn, onOut);

});

// Функция которая отработает при наведении курсора на элемент

function onIn() {

$("div").html("IN");

}

// Функция которая отработает при выходе курсора за элемент

function onOut() {

$("div").html("OUT");

}

</script>

<style>

div {

background: green;

width: 100px;

height: 100px;

text-align: center;

color: white;

}

</style>

</head>

<body>

<div>

Test div

</div>

### MouseMove

<title>Событие mousemove</title>

<script src="jquery-1.10.2.js"></script>

<script>

// mousemove() - функция, устанавливающая обработчик на событие движения курсора по поверхности элемента.

$(function () {

$("div").mousemove(function (e) {

// парметр "e" - объект с информацией о событии.

// pageX, pageY - координаты относительно документа, учитывая прокрутку.

var pageCoord = "( " + e.pageX + ", " + e.pageY + " )";

$("span").text("Page coords " + pageCoord);

});

});

</script>

<style>

div {

background-color: Yellow;

width: 300px;

height: 300px

}

body {

height: 2000px;

}

</style>

</head>

<body>

<span>Coords</span>

<div></div>

<span>Coords</span>

</body>

## EventHadlers

### AddEventListener

<title>Метод addEventListener()</title>

<script>

window.addEventListener('load', function () { // Установка обработичка на событие load.

function show\_msg() {

alert('Hello world');

}

var div = document.getElementById('test');

div.addEventListener('mouseover', show\_msg, false); // Установка обработичка на событие mouseover для элемента c id test.

}, false);

</script>

### RemoveEventListener

<script>

window.addEventListener('load', function () { // Установка обработичка на событие load.

var div = document.getElementById('test');

var del = document.getElementById('del');

function show\_msg() {

alert('Hello world');

}

div.addEventListener('mouseover', show\_msg, false); // Установка обработичка на событие mouseover для дива.

del.addEventListener('click', function(){

div.removeEventListener('mouseover', show\_msg, false); // Удаление обработичка на событие mouseover для дива.

}, false);

}, false);

</script>

### EventRoutes

// В модели DOM Level 2 события распространяются по следующему принципу:

// На этапе ПЕРЕХВАТА событие распространяется от Document вниз до целевого элемента управления.

// Второй этап - СОБЫТИЕ В ЦЕЛЕВОМ УЗЛЕ.

// На третьем этапе событие ВСПЛЫВАЕТ от целевого элемента обратно вверх до корневого элемента.

// последний параметр в методе addEventListener -

// true - обработать событие на этапе ПЕРЕХВАТА

// false - обработать событие на этапе ВСПЛЫВАНИЕ

var isTunnel = true;

window.addEventListener('load', function () {

var e1 = document.getElementById("first");

var e2 = document.getElementById("second");

var e3 = document.getElementById("third");

e1.addEventListener("click",

function () {

this.style.backgroundColor = "green";

alert("Обработчик DIV 1");

},

isTunnel);

e2.addEventListener("click",

function () {

this.style.backgroundColor = "yellow";

alert("Обработчик DIV 2");

},

isTunnel);

e3.addEventListener("click",

function () {

this.style.backgroundColor = "red";

alert("Обработчик DIV 3");

},

isTunnel);

}, false);

</script>

<style>

div {

border: 3px solid gray;

padding: 50px;

background-color: white;

}

</style>

</head>

<body>

Первый

<div id="first">

Второй

<div id="second">

Третий

<div id="third">

</div>

</div>

</div>

### PreventDefault

<title>Метод preventDefault()</title>

<script>

window.addEventListener("load", function () {

document.forms[0].addEventListener("submit", function (e) {

if (document.getElementById("nameInput").value.length == 0)

{

// Если поле ввода пустое - отменить действие по умолчанию, которое связанно с событием.

e.preventDefault();

alert("Введите имя");

}

}, false);

}, false);

</script>

</head>

<body>

<form action="/test.html" method="post">

Имя <input id="nameInput" type="text" name="name"/>

<input type="submit" value="Отправить" />

</form>

### StopPropagation

<title>Метод stopPropogation()</title>

<script>

var isTunnel = false;

window.onload = function () {

var e1 = document.getElementById("first");

var e2 = document.getElementById("second");

var e3 = document.getElementById("third");

e1.addEventListener("click",

// e - параметр хранит объект с информацией о событии.

function (e) {

this.style.backgroundColor = "green";

alert("Обработчик DIV 1");

// прекращение дальнейшего маршрута события.

e.stopPropagation();

},

isTunnel);

e2.addEventListener("click",

function (e) {

this.style.backgroundColor = "yellow";

alert("Обработчик DIV 2");

e.stopPropagation();

},

isTunnel);

e3.addEventListener("click",

function (e) {

this.style.backgroundColor = "red";

alert("Обработчик DIV 3");

e.stopPropagation();

},

isTunnel);

}

</script>

<style>

div {

border: 3px solid gray;

padding: 50px;

background-color: white;

}

</style>

</head>

<body>

Первый

<div id="first">

Второй

<div id="second">

Третий

<div id="third">

</div>

</div>

</div>

### On()/Target/Делегированная обработка событий

<title>Метод on()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод on(событие, обработчик) используется для привязки обработчиков на события к объекту или объектам.

// Он принимает в качестве аругментов событие и функцию обработчик. Предусмотрено добавление дополнительных параметров. В отличии от addEventListener(), который может работать как на "перехвате", так и на всплытии, метод on() работает на "всплытие".

//

$(function () {

$("ul").on("click", function (event) {

$(event.target).css("background", "red");

});

});

</script>

</head>

<body>

<ul>

<li>Текст</li>

<li>Текст</li>

<li>Текст</li>

<li>Текст</li>

<li>Текст</li>

<li>Текст</li>

<li>Текст</li>

<li>Текст</li>

<li>Текст</li>

<li>Текст</li>

</ul>

<img src="DelegateOn.gif" />

</body>

### On() with option

<script src="jquery-1.10.2.js"></script>

<script>

// Метод on(событие, обработчик) используется для привязки обработчиков на события к объекту или объектам.

// Также есть возможность передавать дополнительные данные в виде объектов.

$(function () {

function user\_info(event) {

$('#out').html("Login: " + event.data.login + "<br />" + "Password: " + event.data.password);

}

// Создание и передача дополнительных данных в виде объектов

var access = { login: 'admin', password: '12345' };

$('#button').on('click', access, user\_info);

});

</script>

</head>

<body>

<p id="out"></p>

<input type="button" id="button" value="click me" />

</body

### Off()

<script src="jquery-1.10.2.js"></script>

<script>

// Метод off(событие, обработчик) - удаляет с выбранных элементов страницы обработчики событий, установленные с помощью метода .on().

$(function () {

function show() {

alert('Hello world');

}

function remove() {

$("#first").off('click', show);

}

$("#first").on('click',show);

$("#second").on('click', remove);

});

</script>

</head>

<body>

<input type="button" value="on" id="first" />

<input type="button" value="off" id="second" />

</body>

### One()

<title>Метод one()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод one() - устанавливает обработчик события выбранным элементам страницы. Особенностью метода является то, что обработчик будет вызван не более одного раза, на каждом из элементов.

$(function () {

var count = 0;

function add\_text() {

count++;

$('#out').text('Lorem ipsum ' + count);

}

$('#out').one('mouseover', add\_text);

//$('#out').on('mouseover', add\_text);

});

</script>

<style>

div {

width: 100px;

height: 100px;

background: yellow;

}

</style>

</head>

<body>

<div id="out"></div>

</body>

### Trigger

<title>Метод trigger()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод trigger() - позволяет запустить событие другого элемента управления на странице

var button1;

var button2;

var button1Ctr = 0;

var button2Ctr = 0;

$(function () {

button1 = $("#first");

button2 = $("#second");

button1.click(onButton1Click);

button2.click(onButton2Click);

});

function onButton1Click() {

button1Ctr++;

$("p:first").text("Button 1 clicked " + button1Ctr);

}

function onButton2Click() {

button2Ctr++;

$("p:last").text("Button 2 clicked " + button2Ctr);

// При нажатии на вторую кнопку, также происходит нажатие на первую кнопку.

button1.trigger("click");

}

</script>

</head>

<body>

<input type="button" id="first" value="Button 1" />

<input type="button" id="second" value="Button 2" />

<p>

Button 1 clicked

</p>

<p>

Button 2 clicked

</p>

</body>

# Animation and Effects

## Animation

### Animate

<title>Функция animate()</title>

<script src="jquery-1.10.2.js"></script>

<script>

/\* функция animate() - выполняет заданную пользователем анимацию, с выбранными элементами. Анимация происходит за счет плавного изменения CSS-свойств у элементов. Функция имеет два варианта использования:

1) .animate(properties, [duration], [easing], [callback])

properties — список CSS-свойств, участвующих в анимации и их конечных значений.

duration — продолжительность выполнения анимации. Может быть задана в миллисекундах или строковым значением 'fast' или 'slow' (200 и 600 миллисекунд).

easing — изменение скорости анимации (будет ли она замедляется к концу выполнения или наоборот ускорится).

callback — функция, которая будет вызвана после завершения анимации.

2) .animate(properties, options)

properties — список CSS-свойств, участвующих в анимации и их конечных значений.

options — дополнительные опции. Должны быть представлены объектом, в формате опция:значение. Варианты опций:

duration — продолжительность выполнения анимации (см. выше).

easing — изменение скорости анимации (будет ли она замедляется к концу выполнения или наоборот ускориться).

complete — функция, которая будет вызвана после завершения анимации.

step — функция, которая будет вызвана после каждого шага анимации.

queue — булево значение, указывающее, следует ли помещать текущую анимацию в очередь функций. В случае false, анимация будет запущена сразу же, не вставая в очередь.

specialEasing — позволяет установить разные значения easing, для разных CSS-параметров. Задается объектом, в формате параметр:значение. Опция была добавлена в jQuery 1.4.\*/

$(function () {

$("div").animate(

{

opacity: 1, // меняем значение прозрачности с текущего до 0.25

marginLeft: '500px' // меняем margin-left с текущего до 150px

}, 3000);

});

</script>

<style>

#id1 {

background-color: Green;

width: 100px;

height: 100px;

text-align: center;

color: White;

margin-left:100px;

opacity:0;

}

</style>

</head>

<body>

<div id="id1">

Test div

</div>

</body>

</html>

### AnimateList

<title>Несколько анимаций одновременно</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

$("div")

.animate(

{

opacity: 0.25,

marginLeft: '550px'

}, 5000)

.animate(

{

opacity: 1.0,

marginLeft: '0px'

}, 5000);

// анимации будут выполняться друг за другом

});

</script>

<style>

#id1 {

background-color: Green;

width: 100px;

height: 100px;

text-align: center;

color: White;

}

</style>

</head>

<body>

<div id="id1">

Test div

</div>

</body>

</html>

### Duration

<title>Параметр duration</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

// параметр duration - продолжительность выполнения анимации. Может быть задан в миллисекундах или строковым значением 'fast' или 'slow' (200 и 600 миллисекунд).

$("div").animate(

{

opacity: 0.25,

marginLeft: '550px'

},

"slow");

});

</script>

<style>

#id1 {

background-color: Green;

width: 100px;

height: 100px;

text-align: center;

color: White;

}

</style>

</head>

<body>

<div id="id1">

Test div

</div>

</body>

### Queue

<title>Параметр queue</title>

<script src="jquery-1.10.2.js"></script>

<script>

// параметр queue — булево значение, указывающее, следует ли помещать текущую анимацию в очередь функций.

// В случае false, анимация будет запущена сразу же, не вставая в очередь (функции будут запускаться одновременно).

$(function () {

var anim = {

marginLeft: '150px',

marginTop: '150px',

};

var options = {

queue: false, // queue: fasle - заставляет вторую анимацию выполняться параллельно с первой

queue: true,

duration: 5000,

};

function a()

{alert("dsds");}

$("div")

.animate(anim, options)

.animate(

{

width: '50px',

height: '50px'

}, 5000);

});

</script>

<style>

div {

background-color: Green;

width: 100px; height: 100px;

text-align: center;

color: White;

}

</style>

</head>

<body>

<div>

Test div

</div>

</body>

</html>

### Easing

<title>Параметр easing</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

// параметр ease - имя функции, которая контролирует скорость воспроизведения анимации.

// возможные значения - linear линейная скоротсь анимации, swing - ускорение в начале замедление в конце

$("div").animate(

{

opacity: 0.25,

marginLeft: '650px'

},

2000,

"linear");

});

</script>

<style>

div {

background-color: Green;

width: 100px;

height: 100px;

text-align: center;

color: White

}

</style>

</head>

<body>

<div>

Test div

</div>

</body>

### Animate callback

<title>call-back function</title>

<script src="jquery-1.10.2.js"></script>

<script>

// последний параметр функции animate() - функция, которая будет выполнена после завершения анимации.

$(function () {

$("div").animate(

{

opacity: 0.25,

marginLeft: '150px'

},

5000,

function () { // callback функция, которая будет вызвана по завершению анимации.

alert("animation complete!!!");

});

});

</script>

<style>

div {

background-color: Green;

width: 100px;

height: 100px;

text-align: center;

color: White

}

</style>

</head>

<body>

<div>

Test div

</div>

</body>

### Example

<title>Animate() example</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

var anim\_block = $('#anim\_block');

var options\_for\_div\_over = { width: '300px', height: '200px' }; // Опции стилей при наведении курсора мыши

var options\_for\_div\_out = { width: '100px', height: '21px' }; // Опции силей при отведении курсора мыши

anim\_block.mouseover(function (e) {

anim\_block.animate(options\_for\_div\_over, 400);

});

anim\_block.mouseout(function () {

anim\_block.animate(options\_for\_div\_out, 400);

});

})

</script>

<style>

#anim\_block {

background-color: Green;

width: 100px;

text-align: center;

color: White;

position: absolute;

left: 40px;

top: 40px;

}

</style>

</head>

<body>

<div id="anim\_block">

read more...

</div>

</body>

## Effects

### Show

<title>Метод show()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Show - делает видимым любой элемент если он был скрыт

$(function () {

$("#show1").click(function () {

$("p").show();

})

$("#show2").click(function () {

$("#div1").show("slow"); // значения - slow, normal, fast

//$("#div1").show(7000); // время анимации, заданное в миллисекундах

})

$("#show3").click(function () {

$("#div2").show("normal", Func); // 1 аргумент - скорость анимации 2 аргумент - функция, которая будет вызвана после анимации

})

function Func() {

$("#div2 > span").show();

}

});

</script>

<style>

#div1, #div2 {

background-color: Yellow;

width: 100px;

height: 100px;

display: none;

}

</style>

</head>

<body>

<input id="show1" type="button" value="Show" />

<p style="display: none">

Hidden Text

</p>

<br />

<input id="show2" type="button" value="Show Slow" />

<div id="div1">

</div>

<br />

<input id="show3" type="button" value="Show And Run" />

<div id="div2">

<span style="display: none;">Hidden Text</span>

</div>

</body>

### Hide

<title>Метод hide()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод hide() - делает невидимым любой отображаемый элемент

$(function () {

$("#hide1").click(function () {

$("p").hide();

})

$("#hide2").click(function () {

$("#div1").hide("slow"); // значения - slow, normal, fast

//$("#div1").hide(7000); // время анимации, заданное в миллисекундах

})

$("#hide3").click(function () {

$("#div2").hide("normal", Func); // 1 аргумент - скорость анимации 2 аргумент - функция, которая будет вызвана после анимации

})

function Func() {

alert("Not Visible");

}

});

</script>

<style>

#div1, #div2 {

background-color: Yellow;

width: 100px;

height: 100px;

}

</style>

</head>

<body>

<input id="hide1" type="button" value="Hide" />

<p>

Visible Text

</p>

<br />

<input id="hide2" type="button" value="Hide Slow" />

<div id="div1">

</div>

<br />

<input id="hide3" type="button" value="Hide And Run" />

<div id="div2">

<span>Visible Text</span>

</div>

</body>

### Toggle

<title>Метод toggle()</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

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

// toggle - функция, переключающая режим отображения каждого из элементов набора.

// Если элемент скрыт, то данная функция отображает его (используя метод show);

// Если элемент видимый — скрывает его (используя метод hide).

$("p").toggle(5000);

});

});

</script>

<style>

p {

background-color: green;

color: white;

padding: 10px;

}

</style>

</head>

<body>

<button>

Show/Hide.

</button>

<p>

Paragraph.

</p>

</body>

### Toggle2

<title>Toggle sample</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

$("#button1").click(function () {

$("p").toggle(5000);

});

});

</script>

</head>

<body>

<p style="background-color: red">

Параграф №1.

</p>

<p style="background-color: yellow">

Параграф №2.

</p>

<p style="background-color: green">

Параграф №3.

</p>

<p style="background-color: aqua">

Параграф №4.

</p>

<p style="background-color: gray">

Параграф №5.

</p>

<input type="button" id="button1" value="Show/Hide" />

</body>

### SlideDown

<title>Метод slideDown()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// slideDown - плавное изменение высоты выбранных элементов (высота увеличивается)

$(function () {

$('p').click(function () {

// is(выражение) - проверяет текущий набор элементов на соответствие указанному выражению и возвращает true, если хотя бы один элемент соответствует выражению.

// если первый div не отображается - возвращается значение true

if ($("div").is(":hidden")) {

$("div").slideDown("slow");

} else {

$("div").hide(5000);

}

});

});

</script>

<style>

div {

background: #de9a44;

margin: 3px;

width: 80px;

height: 40px;

display: none;

float: left;

}

</style>

</head>

<body>

<p>Click me!</p>

<div>

</div>

<div>

</div>

<div>

</div>

</body>

</html>

### SlideUp

<title>Метод slideUp</title>

<script src="jquery-1.10.2.js"></script>

<script>

// slideUp - плавное изменение высоты выбранных элементов (высота уменьшается)

$(function () {

$('p').click(function () {

if ($("div").is(":hidden")) {

$("div").show("slow");

} else {

$("div").slideUp();

}

});

});

</script>

### SlideUp2

<title>Effects Slide Up</title>

<script src="jquery-1.10.2.js"></script>

<script>

// slideUp(скорость, функция) - после окончания анимации будет вызывана функция

$(function () {

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

//parent() получение родительского элемента

$(this).parent().slideUp("slow", function () {

$("#msg").text("Кнопка " + $(this).text() + " спрятана");

});

});

});

</script>

<style>

div {

margin: 2px;

}

</style>

</head>

<body>

<div>

<button>

Hide One</button>

<input type="text" value="One" />

</div>

<div>

<button>

Hide Two</button>

<input type="text" value="Two" />

</div>

<div>

<button>

Hide Three</button>

<input type="text" value="Three" />

</div>

<div id="msg">

</div>

</body>

</html>

### SlideToggle

<script src="jquery-1.10.2.js"></script>

<script>

// Метод slideToggle() - переключает видимость всех совпавших элементов набора, используя эффект изменения высоты элементов.

$(function () {

$("#button1").click(function () {

$("p").slideToggle("slow");

});

});

</script>

</head>

<body>

<p style="background-color: red">

Параграф №1.

</p>

<p style="background-color: yellow">

Параграф №2.

</p>

<p style="background-color: green">

Параграф №3.

</p>

<p style="background-color: aqua">

Параграф №4.

</p>

<p style="background-color: gray">

Параграф №5.

</p>

<input type="button" id="button1" value="Slide Down / Slide Up" />

</body>

### FadeIn

<title>Метод fadeIn()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод fadeIn - плавное изменение прозрачности элемента до полной видимости элемента (для работы эффекта элемент должен быть скрыт)

$(function () {

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

// div: hidden: first - первый скрытый div

$("div:hidden:first").fadeIn("slow");

});

});

</script>

<style>

p {

color: red;

cursor: pointer;

}

div {

margin: 3px;

width: 80px;

display: none;

height: 80px;

float: left;

}

div#one {

background: #f00;

}

div#two {

background: #0f0;

}

div#three {

background: #00f;

}

</style>

</head>

<body>

<p>Click here...</p>

<div id="one">

</div>

<div id="two">

</div>

<div id="three">

</div>

</body>

### FadeOut

<title>Метод fadeOut()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод fadeOut - плавно делает элемент невидимым, меняя его прозрачность.

$(function () {

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

$(this).fadeOut("slow");

});

});

</script>

<style>

p {

font-size: 150%;

cursor: pointer;

}

</style>

</head>

<body>

<p>

Click me

</p>

</body>

</html>

### FadeOut

<title>Метод fadeTo()</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

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

// Первый параметр - скорость анимации, второй - прозрачность до которой анимация будет проигрываться

// 1 - элемент полностью непрозрачен 0 - элемент скрыт

$(this).fadeTo("slow", 0.33);

});

});

</script>

</head>

<body>

<p>

Paragraph.

</p>

<p>

Paragraph.

</p>

</body>

</html>

## Queue

### Delay

<title>Effects Show</title>

<script src="jquery-1.10.2.js"></script>

<script>

// delay() - функция, которая останавливает выполнение анимации на установленное время

$(function () {

$("#button1").click(function () {

$("p")

.fadeOut("slow")

.delay(1000)

.fadeIn("slow");

});

});

</script>

</head>

<body>

<p style="position:absolute;margin:auto auto" >Параграф №1</p>

<input type="button" id="button1" value="Push Me!" style="position:absolute;margin-top:30px" />

</body>

</html>

### Queue

<title>Метод queue()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// queue - отображает массив функций, которые должны быть выполненны для указанного элемента. По умолчанию все манипуляции производятся с очередью “fx” – т.е. очередь со всей анимацией

$(function () {

var div = $("div");

function runIt() {

div.show("slow");

div.animate({ left: '+=200' }, 2000);

div.slideToggle(1000);

div.slideToggle("fast");

div.animate({ left: '-=200' }, 1500);

div.hide("slow");

div.show(1200);

div.slideUp("normal", runIt);

}

function showQueueLength() {

var q = div.queue('fx');

$("span").text("Queue length is: " + q.length);

setTimeout(showQueueLength, 100);

}

runIt();

showQueueLength();

});

</script>

</head>

<body>

<span></span>

<div style="width:100px; height:100px; background-color:green; position:absolute"></div>

</body>

### Dequeue

<title>Метод dequeue()</title>

<script src="jquery-1.10.2.js"></script>

<script>

//function(){

// "Тело функции".dequeue() - выполнить тело функции в порядке очереди.

//}

// queue - отображает массив функций, которые должны быть выполненны для указанного элемента. По умолчанию все манипуляции производятся с очередью “fx” – т.е. очередь со всей анимацией

$(function () {

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

$("div")

.animate({ left: "200px" }, 2000)

.animate({ top: "0px" }, 600)

.queue(function () {

$(this).toggleClass("red").dequeue();

})

.animate({ left: "10px", top: "30px" }, 700);

});

});

</script>

<style>

div {

margin: 3px;

width: 50px;

position: absolute;

height: 50px;

left: 10px;

top: 30px;

background-color: yellow;

}

div.red {

background-color: red;

}

</style>

</head>

<body>

<button>Start</button>

<div></div>

</body>

### ClearQueue

<title>clearQueue</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод clearQueue() - удаление всех очередей

$(function () {

var div = $("div");

$("#button1").click(function () {

div.clearQueue();

});

function showQueueLength() {

var q = div.queue("fx");

$("span").text("Queue length is: " + q.length);

setTimeout(showQueueLength, 100);

}

function runIt() {

div.show("slow");

div.animate({ left: '+=200' }, 2000);

div.slideToggle(1000);

div.slideToggle("fast");

div.animate({ left: '-=200' }, 1500);

div.hide("slow");

div.show(1200);

div.slideUp("normal", runIt);

}

runIt();

showQueueLength();

});

</script>

</head>

<body>

<input type="button" id="button1" value="Очистить очередь" />

<span></span>

<div style="width: 100px; height: 100px; background-color: green; position: absolute"></div>

</body>

### Stop

<title>Метод stop()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод stop([clearQueue], [jumpToEnd]) - останавливает выполнение текущей анимации.

// После остановки анимации, следующая в очереди будет запущена автоматически если очередь не пуста.

// Если необходимо остановить всю очередь, то необходимо указать true в параметре clearQueue.

// По умолчанию, элемент останется в том положении, в котором он находился в момент вызова .stop(), обработчик завершения анимации вызван не будет.

// Однако, если указать в параметре jumpToEnd значение true, то после остановки анимации, элемент примет состояние, которое он должен был принять по ее завершению.

// Кроме этого, будет вызван обработчик окончания анимации если он был задан.

$(function () {

$("#go").click(function () {

$(".block").animate({ left: '1000px' }, 2000).animate({ top: '200px' }, 1000);

});

$("#stop").click(function () {

$(".block").stop(); // остановить анимацию на текущем месте (и запустить следующую если она есть)

//$(".block").stop(false, true); // остановить текущую анимацию перейдя в конечное состояние анимации, но не очищать другие запланированные анимации.

//$(".block").stop(true, true); // остановить текущую анимацию перейдя в конечное состояние анимации и очистить очередь

});

$("#back").click(function () {

$(".block").animate({ left: '0px' }, 2000);

});

});

</script>

<style>

div {

position: absolute;

background-color: green;

left: 0px;

top: 30px;

width: 60px;

height: 60px;

margin: 5px;

}

</style>

</head>

<body>

<input type="button" id="go" value="Go" />

<input type="button" id="stop" value="Stop" />

<input type="button" id="back" value="Back" />

<div class="block"></div>

</body>

</html>

# JQueryUI

## Accordion

### Accordion

<title>jQuery UI Accordion</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.accordion.js"></script>

<script>

$(function () {

// Инициализируем элемент управления

$("#accordion").accordion();

});

</script>

</head>

<body>

<!-- Widget - аккордеон -->

<div id="accordion">

<!-- Заголовок - отдельная секция -->

<h3>Section 1</h3>

<!-- div - контент, который отображается при клике на секцию-->

<div>

<p>

Mauris

</p>

</div>

<h3>Section 2</h3>

<div>

<p>

Sed

</p>

</div>

<h3>Section 3</h3>

<div>

<p>

Nam

</p>

<ul>

<li>List item one</li>

<li>List item two</li>

<li>List item three</li>

</ul>

</div>

<h3>Section 4</h3>

<div>

<p>

Cras

</p>

<p>

Suspendisse

</p>

</div>

</div>

</body>

</html>

### FillSpace

<title>jQuery UI Accordion - Fill space</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.resizable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.accordion.js"></script>

<script>

// resizable - разширение библиотеки JQuery, которое позволяет изменять размер элемента при помощи мыши http://jqueryui.com/resizable/. Дополнительно нужно подключить планин jquery.ui.mouse.js

$(function () {

$("#accordion").accordion({

// определение поведения секции по высоте

heightStyle: "fill"

});

// Аккордеон состоит из блочных элементов. По умолчанию его ширина заполняет все доступное пространство по горизонтали.

// Для заполнения вертикального пространства, выделенного в его контейнере, установите свойство heightStyle в "fill"

// и скрипт будет автоматически устанавливать размеры аккордеона по высоте своего родительского контейнера.

$("#accordion-resizer").resizable({

minheight: 140,

minwidth: 200,

resize: function () {

$("#accordion").accordion("refresh");

}

});

});

</script>

<style>

#accordion-resizer {

padding: 10px;

width: 350px;

height: 220px;

}

</style>

</head>

<body>

<h3 class="docs">Начните изменять размер контейнера:</h3>

<div id="accordion-resizer" class="ui-widget-content">

<div id="accordion">

<h3>Section 1</h3>

<div>

<p>Mauris mauris ante, blandit et, ultrices a, suscipit eget, quam. Integer ut neque. Vivamus nisi metus, molestie vel, gravida in, condimentum sit amet, nunc. Nam a nibh. Donec suscipit eros. Nam mi. Proin viverra leo ut odio. Curabitur malesuada. Vestibulum a velit eu ante scelerisque vulputate.</p>

</div>

<h3>Section 2</h3>

<div>

<p>Sed non urna. Donec et ante. Phasellus eu ligula. Vestibulum sit amet 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 3</h3>

<div>

<p>Nam enim risus, molestie et, porta ac, aliquam ac, risus. Quisque lobortis. Phasellus pellentesque purus in massa. Aenean in pede. Phasellus ac libero ac tellus pellentesque semper. Sed ac felis. Sed commodo, magna quis lacinia ornare, quam ante aliquam nisi, eu iaculis leo purus venenatis dui. </p>

<ul>

<li>List item one</li>

<li>List item two</li>

<li>List item three</li>

</ul>

</div>

<h3>Section 4</h3>

<div>

<p>Cras dictum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean lacinia mauris vel est. </p>

<p>Suspendisse eu nisl. Nullam ut libero. Integer dignissim consequat lectus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. </p>

</div>

</div>

</div>

</body>

</html>

### HeightStyle

<script>

$(function () {

$("#accordion").accordion({

// Установите свойство heightStyle: "content". Это позволяет оставить базовую высоту аккордеону (по контенту).

heightStyle: "content"

});

});

</script>

</head

### Icons

<title>jQuery UI Accordion - Customize icons</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.accordion.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script>

$(function () {

// Создаем объект, описывающий набор иконок http://api.jqueryui.com/theming/icons/

var icons = {

header: "ui-icon-circle-arrow-e",

activeHeader: "ui-icon-circle-arrow-s"

};

$("#accordion").accordion({

icons: icons

});

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

// если передается параметр option при вызове методов, подобных accordion - это указывает на то, что мы не создаем новый widget аккордион, а меняем настройки уже существующего

if ($("#accordion").accordion("option", "icons")) {

// отключение иконок

$("#accordion").accordion("option", "icons", null);

} else {

// установка иконок

$("#accordion").accordion("option", "icons", icons);

}

});

});

</script>

</head>

<body>

<div id="accordion">

<h3>Section 1</h3>

<div>

<p>Mauris mauris ante, blandit et, ultrices a, suscipit eget, quam. Integer ut neque. Vivamus nisi metus, molestie vel, gravida in, condimentum sit amet, nunc. Nam a nibh. Donec suscipit eros. Nam mi. Proin viverra leo ut odio. Curabitur malesuada. Vestibulum a velit eu ante scelerisque vulputate.</p>

</div>

<h3>Section 2</h3>

<div>

<p>Sed non urna. Donec et ante. Phasellus eu ligula. Vestibulum sit amet 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 3</h3>

<div>

<p>Nam enim risus, molestie et, porta ac, aliquam ac, risus. Quisque lobortis. Phasellus pellentesque purus in massa. Aenean in pede. Phasellus ac libero ac tellus pellentesque semper. Sed ac felis. Sed commodo, magna quis lacinia ornare, quam ante aliquam nisi, eu iaculis leo purus venenatis dui. </p>

<ul>

<li>List item one</li>

<li>List item two</li>

<li>List item three</li>

</ul>

</div>

<h3>Section 4</h3>

<div>

<p>Cras dictum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean lacinia mauris vel est. </p>

<p>Suspendisse eu nisl. Nullam ut libero. Integer dignissim consequat lectus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. </p>

</div>

</div>

<button id="toggle">Toggle icons</button>

</body>

</html>

### Sortable

<head>

<title>jQuery UI Accordion - Sortable</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.sortable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.accordion.js"></script>

<script>

$(function () {

$("#accordion")

.accordion({

// Указываем порядок вложености секции, если она отлична от заголовка

header: "> div > h3"

}).sortable({ // Устанавливаем возможность сортировки виджета

// Выбираем ось сортировки

axis: "y",

// Указываем элемент управляющий сортировкой

handle: "h3"

});

});

</script>

</head>

<body>

<div id="accordion">

<div class="group">

<h3>Section 1</h3>

<div>

<p>Mauris mauris ante, blandit et, ultrices a, suscipit eget, quam. Integer ut neque. Vivamus nisi metus, molestie vel, gravida in, condimentum sit amet, nunc. Nam a nibh. Donec suscipit eros. Nam mi. Proin viverra leo ut odio. Curabitur malesuada. Vestibulum a velit eu ante scelerisque vulputate.</p>

</div>

</div>

<div class="group">

<h3>Section 2</h3>

<div>

<p>Sed non urna. Donec et ante. Phasellus eu ligula. Vestibulum sit amet 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>

</div>

<div class="group">

<h3>Section 3</h3>

<div>

<p>Nam enim risus, molestie et, porta ac, aliquam ac, risus. Quisque lobortis. Phasellus pellentesque purus in massa. Aenean in pede. Phasellus ac libero ac tellus pellentesque semper. Sed ac felis. Sed commodo, magna quis lacinia ornare, quam ante aliquam nisi, eu iaculis leo purus venenatis dui. </p>

<ul>

<li>List item one</li>

<li>List item two</li>

<li>List item three</li>

</ul>

</div>

</div>

<div class="group">

<h3>Section 4</h3>

<div>

<p>Cras dictum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean lacinia mauris vel est. </p>

<p>Suspendisse eu nisl. Nullam ut libero. Integer dignissim consequat lectus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. </p>

</div>

</div>

</div>

</body>

## Autocomplete

### AutoField

<title>jQuery UI Autocomplete - Default functionality</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.menu.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.autocomplete.js"></script>

<script>

$(function () {

var availableTags = [ // Создаем массив значений для автополя

"ActionScript",

"AppleScript",

"Asp",

"BASIC",

"C",

"C++",

"Clojure",

"COBOL",

"ColdFusion",

"Erlang",

"Fortran",

"Groovy",

"Haskell",

"Java",

"JavaScript",

"Lisp",

"Perl",

"PHP",

"Python",

"Ruby",

"Scala",

"Scheme"

];

$("#tags").autocomplete({ // инициализируем автополе и привязываем к полю источник

source: availableTags

});

});

</script>

</head>

<body>

<div class="ui-widget">

Tags:

<input id="tags" type="text" />

</div>

</body>

### AutoField SelectData

<title>jQuery UI Autocomplete - Custom data and display</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.menu.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.autocomplete.js"></script>

<style>

#project-label {

display: block;

font-weight: bold;

margin-bottom: 1em;

}

#project-icon {

float: left;

height: 32px;

width: 32px;

}

#project-description {

margin: 0;

padding: 0;

}

</style>

<script>

$(function () {

// Создаем массив объектов для автополя

var projects = [

{

value: "jquery",

label: "jQuery",

desc: "the write less, do more, JavaScript library",

icon: "jquery\_32x32.png"

},

{

value: "jquery-ui",

label: "jQuery UI",

desc: "the official user interface library for jQuery",

icon: "jqueryui\_32x32.png"

},

{

value: "sizzlejs",

label: "Sizzle JS",

desc: "a pure-JavaScript CSS selector engine",

icon: "sizzlejs\_32x32.png"

}

];

$("#project").autocomplete({

// Минимальное количесто символов для поиска

minLength: 1,

// Источник

source: projects,

// Настройка выделения варианта выбора при наведении мыши

focus: function (event, ui) {

$("#project").val(ui.item.label);

return false;

},

// Настройка отображения каждого элемента списка вариантов

select: function (event, ui) {

$("#project").val(ui.item.label);

$("#project-id").val(ui.item.value);

$("#project-description").html(ui.item.desc);

$("#project-icon").attr("src", "images/" + ui.item.icon);

return false;

}

})

// \_renderItem - контролирует отрисовку элементов в выпадающем списке. Метод должен создать <li> добавить его в <ul> и вернуть созданный элемент.

.data("autocomplete").\_renderItem = function (ul, item) {

return $("<li>")

.data("item.autocomplete", item) // указываем информацию, которая будет храниться в каждом элементе списка

.append("<a>" + item.label + "<br>" + item.desc + "</a>")

.appendTo(ul);

};

});

</script>

</head>

<body>

<div id="project-label">Select a project (type "j" for a start):</div>

<img id="project-icon" src="images/transparent\_1x1.png" class="ui-state-default" alt="">

<input id="project" type="text">

<input type="hidden" id="project-id">

<p id="project-description"></p>

</body>

</html>

## Button

### Button()

<title>jQuery UI Button - Default functionality</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script>

$(function () {

$("input[type=submit], a, button") // Находим элементы

.button() // Инициализируем виджеты

.click(function (event) {

alert("Hello");

event.preventDefault(); // Отменяем действие по умолчанию

});

});

</script>

</head>

<body>

<button>A button element</button>

<input type="submit" value="A submit button">

<a href="#">An anchor</a>

<div>

<p>Examples of the markup that can be used for buttons: A button element, an input of type submit and an anchor.</p>

</div>

</body>

</html>

### CheckBox toggle

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script>

$(function () {

$("#check").button(); // если для элемента checkbox вызвать метод button - создастся toggle button

});

</script>

<style>

#format {

margin-top: 16px;

}

</style>

</head>

<body>

<input type="checkbox" id="check" /><label for="check">Toggle</label>

<div>

<p>Checkbox стилизован как toggle button. Элемент label используется как текст для кнопки</p>

</div>

</body>

### Radio

<title>jQuery UI Button</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script>

$(function () {

$("#radio").buttonset(); // buttonset - виджет для группировки радиокнопок

});

</script>

</head>

<body>

<form>

<div id="radio">

<input type="radio" id="radio1" name="radio" /><label for="radio1">Choice 1</label>

<input type="radio" id="radio2" name="radio" checked="checked" /><label for="radio2">Choice 2</label>

<input type="radio" id="radio3" name="radio" /><label for="radio3">Choice 3</label>

</div>

</form>

<div>

<p>Набор из трех элементов радио-кнопок превращается в button set.</p>

</div>

</body>

</html>

### Icons

<!DOCTYPE html>

<html>

<head>

<title>jQuery UI Button - Icons</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script>

// primary - иконка на кнопке слева

// secondary - иконка на кнопке справа

// text: false - не отображать текст на кнопке

$(function () {

$("button:first").button({

icons: {

primary: "ui-icon-locked"

},

text: false

}).next().button({

icons: {

primary: "ui-icon-locked"

}

}).next().button({

icons: {

primary: "ui-icon-gear",

secondary: "ui-icon-triangle-1-s"

}

}).next().button({

icons: {

primary: "ui-icon-gear",

secondary: "ui-icon-triangle-1-s"

},

text: false

});

});

</script>

</head>

<body>

<button>Button with icon only</button>

<button>Button with icon on the left</button>

<button>Button with two icons</button>

<button>Button with two icons and no text</button>

</body>

</html>

## DatePicker

### Datetimepicker()

<!DOCTYPE html>

<html>

<head>

<title>jQuery UI Datepicker - Default functionality</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.datepicker.js"></script>

<script>

$(function () {

// Инициализация виджета datapicker

$("#datepicker").datepicker();

});

</script>

</head>

<body>

<p>

Date:

<input type="text" id="datepicker">

</p>

<div class="demo-description">

<p>

Widget datepicker по умолчанию привязывается к элементу input.

Когда input получает фокус (click или tab key) открывается небольшой календарь.

</p>

</div>

</body>

</html>

### Datepicker icon

<title>jQuery UI Datepicker - Icon trigger</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.datepicker.js"></script>

<script>

$(function () {

$("#datepicker").datepicker({

// настройки для виджета datepicker

showOn: "button", // Отображать при нажатии на кнопку. Возможные значения (focus, button, both)

buttonImage: "images/calendar.gif", // Путь к картинке, которая будет отображаться на кнопке

buttonImageOnly: true // true - не отображать кнопку, отображать только изображение. Работает только тогда, когда установлен параметр buttonImage

});

});

</script>

</head>

<body>

<p>

Date:

<input type="text" id="datepicker">

</p>

</body>

</html>

### Min-max datepicker

<title>jQuery UI Datepicker - Restrict date range</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.datepicker.js"></script>

<script>

$(function () {

$("#datepicker").datepicker(

{

minDate: -20, // минимальная дата

maxDate: "+1M +10D" // максимальная дата (+ 1 месяц и 10 дней)

});

});

</script>

</head>

<body>

<p>

Date:

<input type="text" id="datepicker">

</p>

</body>

</html>

### Multiple month

<title>jQuery UI Datepicker - Display multiple months</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.datepicker.js"></script>

<script>

$(function () {

$("#datepicker").datepicker({

numberOfMonths: 3, // количество месяцев, которые будут отображаться за раз.

showButtonPanel: true // true - отображать панель с кнопками

});

});

</script>

</head>

<body>

<p>

Date:

<input type="text" id="datepicker">

</p>

</body>

</html>

### DateFormat

<html>

<head>

<title>jQuery UI Datepicker - Format date</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.datepicker.js"></script>

<script>

$(function () {

$("#datepicker").datepicker();

$("#format").change(function () {

$("#datepicker").datepicker("option", "dateFormat", $(this).val()); // меняем формат даты в соответствии со значением из поля ввода #format

});

});

</script>

</head>

<body>

<p>

Date:

<input type="text" id="datepicker" size="30" />

</p>

<p>

Format options:<br />

<select id="format">

<option value="mm/dd/yy">Default - mm/dd/yy</option>

<option value="yy-mm-dd">ISO 8601 - yy-mm-dd</option>

<option value="d M, y">Short - d M, y</option>

<option value="d MM, y">Medium - d MM, y</option>

<option value="DD, d MM, yy">Full - DD, d MM, yy</option>

<option value="'day' d 'of' MM 'in the year' yy">With text - 'day' d 'of' MM 'in the year' yy</option>

</select>

</p>

</body>

</html>

### Date range

<title>jQuery UI Datepicker - Select a Date Range</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.datepicker.js"></script>

<script>

$(function () {

$("#from").datepicker({

defaultDate: "+1w", // дата по умолчанию + 1 неделя

changeMonth: true, // есть возможность менять месяц с помощью выпадающего списка

// при смене выбранной даты находим второй datepicker и устанавливаем минимальную дату, равную выбранной

onSelect: function (selectedDate) {

$("#to").datepicker("option", "minDate", selectedDate);

}

});

$("#to").datepicker({

defaultDate: "+1w",

changeMonth: true,

// при смене выбранной даты находим первый datepicker и устанавливаем максимальную дату, равную выбранной

onSelect: function (selectedDate) {

$("#from").datepicker("option", "maxDate", selectedDate);

}

});

});

</script>

</head>

<body>

<label for="from">From</label>

<input type="text" id="from" name="from" />

<label for="to">to</label>

<input type="text" id="to" name="to" />

<div>

<p>Введите диапазон дат</p>

</div>

</body>

</html>

### Inline

<title>jQuery UI Datepicker - Display inline</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.datepicker.js"></script>

<script>

$(function () {

// если datepicker вызвать на элементе div, то календарь будет встроенный в страницу.

$("#datepicker").datepicker();

});

</script>

</head>

<body>

Date:

<div id="datepicker"></div>

</body>

</html

### Datepicker animation

<title>jQuery UI Datepicker - Animations</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-blind.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-bounce.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-clip.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-drop.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-fold.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-slide.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.datepicker.js"></script>

<script>

$(function () {

$("#datepicker").datepicker();

$("#anim").change(function () {

// смена анимации для datepicker

$("#datepicker").datepicker("option", "showAnim", $(this).val());

});

});

</script>

</head>

<body>

<p>Date:

<input type="text" id="datepicker" size="30" /></p>

<p>

Animations:<br />

<select id="anim">

<option value="show">Show (default)</option>

<option value="slideDown">Slide down</option>

<option value="fadeIn">Fade in</option>

<option value="blind">Blind (UI Effect)</option>

<option value="bounce">Bounce (UI Effect)</option>

<option value="clip">Clip (UI Effect)</option>

<option value="drop">Drop (UI Effect)</option>

<option value="fold">Fold (UI Effect)</option>

<option value="slide">Slide (UI Effect)</option>

<option value="">None</option>

</select>

</p>

<div>

<p>Для того, чтобы работали дополнительные эффекты, необходимо подключить соответствующую библиотеку UI effect</p>

</div>

</body>

</html>

### Localization

<script src="../jquery-ui-1.10.3/ui/i18n/jquery.ui.datepicker-zh-CN.js"></script>

<script src="../jquery-ui-1.10.3/ui/i18n/jquery.ui.datepicker-zh-HK.js"></script>

<script src="../jquery-ui-1.10.3/ui/i18n/jquery.ui.datepicker-zh-TW.js"></script>

<script>

$(function () {

// смена настроек для всех datepicker

$.datepicker.setDefaults($.datepicker.regional[""]);

$.datepicker.setDefaults()

// смена локали для "datepicer

$("#datepicker").datepicker($.datepicker.regional["fr"]);

$("#locale").change(function () {

// смена локали при выборе нового значения в выпадающем списке

$("#datepicker").datepicker("option",

$.datepicker.regional[$(this).val()]);

});

});

</script>

</head>

<body>

<p>

Date:

<input type="text" id="datepicker" />&nbsp;

<select id="locale">

<option value="af">Afrikaans</option>

<option value="sq">Albanian (Gjuha shqipe)</option>

<option value="ar-DZ">Algerian Arabic</option>

<option value="ar">Arabic (&#8235;(&#1604;&#1593;&#1585;&#1576;&#1610;</option>

<option value="hy">Armenian (&#1344;&#1377;&#1397;&#1381;&#1408;&#1381;&#1398;)</option>

<option value="az">Azerbaijani (Az&#601;rbaycan dili)</option>

<option value="eu">Basque (Euskara)</option>

<option value="bs">Bosnian (Bosanski)</option>

<option value="bg">Bulgarian (&#1073;&#1098;&#1083;&#1075;&#1072;&#1088;&#1089;&#1082;&#1080; &#1077;&#1079;&#1080;&#1082;)</option>

<option value="ca">Catalan (Catal&agrave;)</option>

<option value="zh-HK">Chinese Hong Kong (&#32321;&#39636;&#20013;&#25991;)</option>

## Dialog

### Dialog

<title>jQuery UI Dialog - Default functionality</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.draggable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.resizable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.dialog.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script>

$(function () {

// создание диалогового окна при загрузке документа

$("#dialog").dialog();

});

</script>

</head>

<body>

<div id="dialog" title="Basic dialog">

<p>Это стандартное диалоговое окно, которое можно передвинуть, поменять размер и закрыть с помощью иконки 'x'.</p>

</div>

</body

### Modal

<title>jQuery UI Dialog - Basic modal</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.draggable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.resizable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.dialog.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script>

$(function () {

$("#dialog-modal").dialog({

height: 140, // определение высоты окна

modal: true // окно будет модальным - доступ к элементам, которые находяться под окном, будет открыт только полсе закрытия самого окна.

});

});

</script>

</head>

<body>

<div id="dialog-modal" title="Basic modal dialog">

<p>При отображении модального диалогового окна страница затемняется.</p>

</div>

</body>

</html>

<title>jQuery UI Dialog - Basic modal</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.draggable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.resizable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.dialog.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script>

$(function () {

$("#dialog-modal").dialog({

height: 140, // определение высоты окна

modal: true // окно будет модальным - доступ к элементам, которые находяться под окном, будет открыт только полсе закрытия самого окна.

});

});

</script>

</head>

<body>

<div id="dialog-modal" title="Basic modal dialog">

<p>При отображении модального диалогового окна страница затемняется.</p>

</div>

</body>

</html>

### Modal message

<title>jQuery UI Dialog - Modal message</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/external/jquery.bgiframe-2.1.2.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.draggable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.resizable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.dialog.js"></script>

<script>

$(function () {

$("#dialog-message").dialog({

modal: true,

// определение кнопок, которые будут отображаться в конце окна

buttons: {

Ok: function () {

// Закрытие текущего диалогового окна

$(this).dialog("close");

}

}

});

});

</script>

</head>

<body>

<div id="dialog-message" title="Download complete">

<p>

<span class="ui-icon ui-icon-circle-check" style="float: left; margin: 0 7px 50px 0;"></span>

Вы успешно скачали файл.

</p>

</div>

</body>

### Modal confirmation

<title>jQuery UI Dialog - Modal confirmation</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/external/jquery.bgiframe-2.1.2.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.draggable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.dialog.js"></script>

<script>

$(function () {

$("#dialog-confirm").dialog({

resizable: false,

height: 270,

width: 400,

modal: true,

buttons: {

// определение первой кнопки

"Удалить все элементы": function () {

$(this).dialog("close");

},

// определение второй кнопки

Cancel: function () {

$(this).dialog("close");

}

}

});

});

</script>

</head>

<body>

<div id="dialog-confirm" title="Empty the recycle bin?">

<p><span class="ui-icon ui-icon-alert" style="float: left; margin: 0 7px 20px 0;"></span>Элементы будут безвозвратно удалены. Вы хотите продолжить?</p>

</div>

</body>

</html>

### Dialog animation

<title>jQuery UI Dialog - Animation</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.draggable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.resizable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.dialog.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-blind.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-explode.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.button.js"></script>

<script>

// определение скорости по умолчанию для анимации

$.fx.speeds.\_default = 5000;

$(function () {

$("#dialog").dialog({

autoOpen: false, // диалоговое окно будет открыто при вызове метода dialog("open")

show: "blind", // эффект для отображения окна

hide: "explode" // эффект для скрытия окна

});

$("#opener").click(function () {

$("#dialog").dialog("open"); // открытие диалогового окна

return false;

});

});

</script>

</head>

<body>

<div id="dialog" title="Basic dialog">

<p>This is an animated dialog which is useful for displaying information. The dialog window can be moved, resized and closed with the 'x' icon.</p>

</div>

<button id="opener">Open Dialog</button>

</body>

</html>

## Menu

### Menu

<title>jQuery UI Menu - Default functionality</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.menu.js"></script>

<script>

$(function () {

$("#menu").menu();

});

</script>

</head>

<body>

<ul id="menu">

<li><a href="#Aberdeen">Aberdeen</a></li>

<li><a href="#Ada">Ada</a></li>

<li><a href="#Adamsville">Adamsville</a></li>

<li><a href="#Addyston">Addyston</a></li>

<li><a href="#Adelphi">Adelphi</a></li>

<li><a href="#Adena">Adena</a></li>

<li><a href="#Adrian">Adrian</a></li>

<li><a href="#Akron">Akron</a></li>

<li><a href="#Albany">Albany</a></li>

<li><a href="#Alexandria">Alexandria</a></li>

<li><a href="#Alger">Alger</a></li>

<li><a href="#Alledonia">Alledonia</a></li>

<li><a href="#Alliance">Alliance</a></li>

<li><a href="#Alpha">Alpha</a></li>

<li><a href="#Alvada">Alvada</a></li>

</ul>

<div class="demo-description">

<p>Меню со стандартной настройкой. К меню добавляется оформление и поддержка клавиатурных команд. Попробуйте использовать клавишу Tab и стрелки.</p>

</div>

</body>

</html>

### Navigation menu

<title>jQuery UI Menu - Navigation Menu</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.menu.js"></script>

<script>

$(function () {

// Создание меню

$("#menu").menu({

// определение обработчика на смену выбранного пункта меню

select: function (event, ui) {

// получение первой ссылки в элементе по которому произвели нажатие

var link = ui.item.children("a:first");

// если при нажатии на ссылку была зажата клавиша ctrl или shift или у элемента присутствует атрибут target - ничего не делаем

if (link.attr("target") || event.metaKey || event.shiftKey || event.ctrlKey) {

return;

}

// производим переход по выбранному адресу

location.href = link.attr("href");

}

});

});

</script>

<style>

.ui-menu {

width: 200px;

margin-bottom: 2em;

}

</style>

</head>

<body>

<ul id="menu">

<li><a href="?Aberdeen">Aberdeen</a></li>

<li><a href="?Ada">Ada</a></li>

<li><a href="?Adamsville">Adamsville</a></li>

<li><a href="?Addyston">Addyston</a></li>

<li>

<a href="?Delphi">Delphi</a>

<ul>

<li><a href="?Ada">Ada</a></li>

<li><a href="?Saarland">Saarland</a></li>

<li><a href="?Salzburg">Salzburg</a></li>

</ul>

</li>

<li><a href="?Saarland">Saarland</a></li>

<li>

<a href="?Salzburg">Salzburg</a>

<ul>

<li>

<a href="?Delphi">Delphi</a>

<ul>

<li><a href="?Ada">Ada</a></li>

<li><a href="?Saarland">Saarland</a></li>

<li><a href="?Salzburg">Salzburg</a></li>

</ul>

</li>

<li>

<a href="?Delphi">Delphi</a>

<ul>

<li><a href="?Ada">Ada</a></li>

<li><a href="?Saarland">Saarland</a></li>

<li><a href="?Salzburg">Salzburg</a></li>

</ul>

</li>

<li><a href="?Perch">Perch</a></li>

</ul>

</li>

</ul>

</body>

</html>

## Progressbar

### Progressbar()

<head>

<title>jQuery UI Progressbar - Default functionality</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.progressbar.js"></script>

<script>

$(function () {

// создание progressbar

$("#progressbar").progressbar({

// определение текущего состояния элемента

value: 37

});

});

</script>

</head>

<body>

<div id="progressbar" ></div>

</body>

</html>

### Resizeable

<!DOCTYPE html>

<html>

<head>

<title>jQuery UI Progressbar - Resizable</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.progressbar.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.resizable.js"></script>

<script>

$(function () {

$("#progressbar").progressbar({

value: 37

});

// определение контейнера, с помощью которого можно будет менять размер содержимого дочерних элементов

$("#progressbarWrapper").resizable();

});

</script>

</head>

<body>

<div id="progressbarWrapper" style="height: 30px;" class="ui-widget-default">

<div id="progressbar" style="height: 100%;"></div>

</div>

<div class="demo-description">

<p>Размеры progress bar заданы в процентах для того, чтобы элемент полностью заполнял свой контейнер.</p>

</div>

</body>

</html>

### Progressbar animation

<head>

<title>jQuery UI Progressbar - Animated</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.progressbar.js"></script>

<script>

$(function () {

$("#progressbar").progressbar({

value: 59

});

});

</script>

<style>

.ui-progressbar .ui-progressbar-value {

background-image: url(images/pbar-ani.gif);

}

</style>

</head>

<body>

<div id="progressbar" style="height: 22px"></div>

<div>

<p>

Для того, чтобы анимировать значение progress bar, необходимо установить свойство

<code>background-image</code>

для класса

<code>.ui-progressbar-value</code>

</p>

</div>

</body>

</html>

## Slider

### Slider()

<title>jQuery UI Slider - Default functionality</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.slider.js"></script>

<script>

$(function () {

// создание слайдера

$("#slider").slider();

});

</script>

</head>

<body>

<div id="slider"></div>

<div>

<p>По умолчанию слайдер имеет горизонтальную ориентацию. Позиция каретки может быть изменена с помощью мыши или клавиш</p>

</div>

</body>

</html>

### Steps

<meta charset="utf-8">

<title>jQuery UI Slider - Snap to increments</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.slider.js"></script>

<script>

$(function () {

$("#slider").slider({

// определение настроек

value: 100,

min: 0,

max: 500,

step: 50,

// обработка смены позиции

slide: function (event, ui) {

$("#amount").val("$" + ui.value);

}

});

$("#amount").val("$" + $("#slider").slider("value"));

});

</script>

</head>

<body>

<p>

<label for="amount">Donation amount ($50 increments):</label>

<input type="text" id="amount" style="border: 0; color: #f6931f; font-weight: bold;" />

</p>

<div id="slider"></div>

</body>

</html>

### Range

<title>jQuery UI Slider - Range slider</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.slider.js"></script>

<link rel="stylesheet" href="../demos.css">

<script>

$(function () {

$("#slider-range").slider({

range: true, // данный слайдер будет использоваться для указания диапазона

min: 0,

max: 500,

// стартовые значения: первая каретка на позиции 75, вторая - 300

values: [75, 300],

slide: function (event, ui) {

$("#amount").val("$" + ui.values[0] + " - $" + ui.values[1]);

}

});

// .slider("values", 0) - получение позиции соответствующей каретки

$("#amount").val("$" + $("#slider-range").slider("values", 0) +

" - $" + $("#slider-range").slider("values", 1));

});

</script>

</head>

<body>

<p>

Price range:

<input type="text" id="amount" style="border: 0; color: #f6931f; font-weight: bold;" />

</p>

<div id="slider-range"></div>

</body>

</html>

### Vertical slider

<title>jQuery UI Slider - Vertical slider</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.slider.js"></script>

<script>

$(function () {

$("#slider-vertical").slider({

orientation: "vertical", // создание вертикального слайдера

range: "min",

min: 0,

max: 100,

value: 60,

slide: function (event, ui) {

$("#amount").val(ui.value);

}

});

$("#amount").val($("#slider-vertical").slider("value"));

});

</script>

</head>

<body>

<p>

<label for="amount">Volume:</label>

<input type="text" id="amount" style="border: 0; color: #f6931f; font-weight: bold;" />

</p>

<div id="slider-vertical" style="height: 200px;"></div>

</body>

</html>

## Tabs

### Tabs()

<title>jQuery UI Tabs - Default functionality</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.tabs.js"></script>

<script>

$(function () {

$("#tabs").tabs();

});

</script>

</head>

<body>

<div id="tabs">

<!--Вкладки-->

<ul>

<li><a href="#tabs-1">Nunc tincidunt</a></li>

<li><a href="#tabs-2">Proin dolor</a></li>

<li><a href="#tabs-3">Aenean lacinia</a></li>

</ul>

<!--Содержимое вкладок-->

<div id="tabs-1">

<p>Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu. Donec sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. Mauris dapibus lacus auctor risus. Aenean tempor ullamcorper leo. Vivamus sed magna quis ligula eleifend adipiscing. Duis orci. Aliquam sodales tortor vitae ipsum. Aliquam nulla. Duis aliquam molestie erat. Ut et mauris vel pede varius sollicitudin. Sed ut dolor nec orci tincidunt interdum. Phasellus ipsum. Nunc tristique tempus lectus.</p>

</div>

<div id="tabs-2">

<p>Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, ut pharetra massa metus id nunc. Duis scelerisque molestie turpis. Sed fringilla, massa eget luctus malesuada, metus eros molestie lectus, ut tempus eros massa ut dolor. Aenean aliquet fringilla sem. Suspendisse sed ligula in ligula suscipit aliquam. Praesent in eros vestibulum mi adipiscing adipiscing. Morbi facilisis. Curabitur ornare consequat nunc. Aenean vel metus. Ut posuere viverra nulla. Aliquam erat volutpat. Pellentesque convallis. Maecenas feugiat, tellus pellentesque pretium posuere, felis lorem euismod felis, eu ornare leo nisi vel felis. Mauris consectetur tortor et purus.</p>

</div>

<div id="tabs-3">

<p>Mauris eleifend est et turpis. Duis id erat. Suspendisse potenti. Aliquam vulputate, pede vel vehicula accumsan, mi neque rutrum erat, eu congue orci lorem eget lorem. Vestibulum non ante. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Fusce sodales. Quisque eu urna vel enim commodo pellentesque. Praesent eu risus hendrerit ligula tempus pretium. Curabitur lorem enim, pretium nec, feugiat nec, luctus a, lacus.</p>

<p>Duis cursus. Maecenas ligula eros, blandit nec, pharetra at, semper at, magna. Nullam ac lacus. Nulla facilisi. Praesent viverra justo vitae neque. Praesent blandit adipiscing velit. Suspendisse potenti. Donec mattis, pede vel pharetra blandit, magna ligula faucibus eros, id euismod lacus dolor eget odio. Nam scelerisque. Donec non libero sed nulla mattis commodo. Ut sagittis. Donec nisi lectus, feugiat porttitor, tempor ac, tempor vitae, pede. Aenean vehicula velit eu tellus interdum rutrum. Maecenas commodo. Pellentesque nec elit. Fusce in lacus. Vivamus a libero vitae lectus hendrerit hendrerit.</p>

</div>

</div>

</body>

</html>

### Mouseover

<title>jQuery UI Tabs - Open on mouseover</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.tabs.js"></script>

<link rel="stylesheet" href="../demos.css">

<script>

$(function () {

// определение на какое событие должно происходить переключение вкладок

$("#tabs").tabs({

event: "mouseover" // по умолчанию "click"

});

});

</script>

</head>

<body>

<div id="tabs">

<ul>

<li><a href="#tabs-1">Nunc tincidunt</a></li>

<li><a href="#tabs-2">Proin dolor</a></li>

<li><a href="#tabs-3">Aenean lacinia</a></li>

</ul>

<div id="tabs-1">

<p>Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu. Donec sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. Mauris dapibus lacus auctor risus. Aenean tempor ullamcorper leo. Vivamus sed magna quis ligula eleifend adipiscing. Duis orci. Aliquam sodales tortor vitae ipsum. Aliquam nulla. Duis aliquam molestie erat. Ut et mauris vel pede varius sollicitudin. Sed ut dolor nec orci tincidunt interdum. Phasellus ipsum. Nunc tristique tempus lectus.</p>

</div>

<div id="tabs-2">

<p>Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, ut pharetra massa metus id nunc. Duis scelerisque molestie turpis. Sed fringilla, massa eget luctus malesuada, metus eros molestie lectus, ut tempus eros massa ut dolor. Aenean aliquet fringilla sem. Suspendisse sed ligula in ligula suscipit aliquam. Praesent in eros vestibulum mi adipiscing adipiscing. Morbi facilisis. Curabitur ornare consequat nunc. Aenean vel metus. Ut posuere viverra nulla. Aliquam erat volutpat. Pellentesque convallis. Maecenas feugiat, tellus pellentesque pretium posuere, felis lorem euismod felis, eu ornare leo nisi vel felis. Mauris consectetur tortor et purus.</p>

</div>

<div id="tabs-3">

<p>Mauris eleifend est et turpis. Duis id erat. Suspendisse potenti. Aliquam vulputate, pede vel vehicula accumsan, mi neque rutrum erat, eu congue orci lorem eget lorem. Vestibulum non ante. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Fusce sodales. Quisque eu urna vel enim commodo pellentesque. Praesent eu risus hendrerit ligula tempus pretium. Curabitur lorem enim, pretium nec, feugiat nec, luctus a, lacus.</p>

<p>Duis cursus. Maecenas ligula eros, blandit nec, pharetra at, semper at, magna. Nullam ac lacus. Nulla facilisi. Praesent viverra justo vitae neque. Praesent blandit adipiscing velit. Suspendisse potenti. Donec mattis, pede vel pharetra blandit, magna ligula faucibus eros, id euismod lacus dolor eget odio. Nam scelerisque. Donec non libero sed nulla mattis commodo. Ut sagittis. Donec nisi lectus, feugiat porttitor, tempor ac, tempor vitae, pede. Aenean vehicula velit eu tellus interdum rutrum. Maecenas commodo. Pellentesque nec elit. Fusce in lacus. Vivamus a libero vitae lectus hendrerit hendrerit.</p>

</div>

</div>

</body>

</html>

### Sortable

<title>jQuery UI Tabs - Sortable</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.mouse.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.sortable.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.tabs.js"></script>

<script>

$(function () {

var tabs = $("#tabs").tabs();

// sortable - преобразование выбранных элементов в элементы, которые поддерживают сортировку

tabs.find(".ui-tabs-nav").sortable({

axis: "x",

stop: function () {

// после завершения сортировки обновить контрол

tabs.tabs("refresh");

}

});

});

</script>

</head>

<body>

<div id="tabs">

<ul>

<li><a href="#tabs-1">Nunc tincidunt</a></li>

<li><a href="#tabs-2">Proin dolor</a></li>

<li><a href="#tabs-3">Aenean lacinia</a></li>

</ul>

<div id="tabs-1">

<p>Proin elit arcu, rutrum commodo, vehicula tempus, commodo a, risus. Curabitur nec arcu. Donec sollicitudin mi sit amet mauris. Nam elementum quam ullamcorper ante. Etiam aliquet massa et lorem. Mauris dapibus lacus auctor risus. Aenean tempor ullamcorper leo. Vivamus sed magna quis ligula eleifend adipiscing. Duis orci. Aliquam sodales tortor vitae ipsum. Aliquam nulla. Duis aliquam molestie erat. Ut et mauris vel pede varius sollicitudin. Sed ut dolor nec orci tincidunt interdum. Phasellus ipsum. Nunc tristique tempus lectus.</p>

</div>

<div id="tabs-2">

<p>Morbi tincidunt, dui sit amet facilisis feugiat, odio metus gravida ante, ut pharetra massa metus id nunc. Duis scelerisque molestie turpis. Sed fringilla, massa eget luctus malesuada, metus eros molestie lectus, ut tempus eros massa ut dolor. Aenean aliquet fringilla sem. Suspendisse sed ligula in ligula suscipit aliquam. Praesent in eros vestibulum mi adipiscing adipiscing. Morbi facilisis. Curabitur ornare consequat nunc. Aenean vel metus. Ut posuere viverra nulla. Aliquam erat volutpat. Pellentesque convallis. Maecenas feugiat, tellus pellentesque pretium posuere, felis lorem euismod felis, eu ornare leo nisi vel felis. Mauris consectetur tortor et purus.</p>

</div>

<div id="tabs-3">

<p>Mauris eleifend est et turpis. Duis id erat. Suspendisse potenti. Aliquam vulputate, pede vel vehicula accumsan, mi neque rutrum erat, eu congue orci lorem eget lorem. Vestibulum non ante. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Fusce sodales. Quisque eu urna vel enim commodo pellentesque. Praesent eu risus hendrerit ligula tempus pretium. Curabitur lorem enim, pretium nec, feugiat nec, luctus a, lacus.</p>

<p>Duis cursus. Maecenas ligula eros, blandit nec, pharetra at, semper at, magna. Nullam ac lacus. Nulla facilisi. Praesent viverra justo vitae neque. Praesent blandit adipiscing velit. Suspendisse potenti. Donec mattis, pede vel pharetra blandit, magna ligula faucibus eros, id euismod lacus dolor eget odio. Nam scelerisque. Donec non libero sed nulla mattis commodo. Ut sagittis. Donec nisi lectus, feugiat porttitor, tempor ac, tempor vitae, pede. Aenean vehicula velit eu tellus interdum rutrum. Maecenas commodo. Pellentesque nec elit. Fusce in lacus. Vivamus a libero vitae lectus hendrerit hendrerit.</p>

</div>

</div>

</body>

</html>

## Tooltips

### Tooltip()

<title>jQuery UI Tooltip - Default functionality</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.tooltip.js"></script>

<script>

$(function () {

// полсе вызова метоад tooltip значения атрибутов title будут отображаться в всплывающей подсказке

$(document).tooltip();

});

</script>

</head>

<body>

<p>

Ваш возраст: <input id="age" title="Укажите, пожалуйста, Ваш возраст">

</p>

</body>

</html>

### Tracking

<title>jQuery UI Tooltip - Track the mouse</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.core.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.widget.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.position.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.tooltip.js"></script>

<script>

$(function () {

$(document).tooltip({

track: true

});

});

</script>

</head>

<body>

<p>

Ваш возраст:<input id="age" title="Укажите, пожалуйста, Ваш возраст"></p>

</body>

</html>

## Effects

<title>jQuery UI Effects - Effect demo</title>

<link rel="stylesheet" href="../jquery-ui-1.10.3/themes/base/jquery.ui.all.css">

<script src="../jquery-ui-1.10.3/jquery-1.9.1.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-blind.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-bounce.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-clip.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-drop.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-explode.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-fade.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-fold.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-highlight.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-pulsate.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-scale.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-shake.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-slide.js"></script>

<script src="../jquery-ui-1.10.3/ui/jquery.ui.effect-transfer.js"></script>

<script>

$(function () {

function runEffect() {

// чтение имени эффекта

var selectedEffect = $("#effectTypes").val();

var options = {};

if (selectedEffect === "scale") {

options = { percent: 0 };

} else if (selectedEffect === "transfer") {

options = { to: "#button", className: "ui-effects-transfer" };

} else if (selectedEffect === "size") {

options = { to: { width: 200, height: 60 } };

}

// применение эффекта

$("#effect").effect(selectedEffect, options, 500, callback);

};

// функция, которая вызывается по завершению эффекта

function callback() {

setTimeout(function () {

$("#effect").removeAttr("style").hide().fadeIn();

}, 1000);

};

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

runEffect();

return false;

});

});

</script>

<style>

.toggler {

width: 500px;

height: 200px;

position: relative;

}

#button {

padding: .5em 1em;

text-decoration: none;

}

#effect {

width: 240px;

height: 135px;

padding: 0.4em;

position: relative;

}

#effect h3 {

margin: 0;

padding: 0.4em;

text-align: center;

}

.ui-effects-transfer {

border: 2px dotted gray;

}

</style>

</head>

<body>

<div class="toggler">

<div id="effect" class="ui-widget-content ui-corner-all">

<h3 class="ui-widget-header ui-corner-all">Effect</h3>

<p>

Etiam libero neque, luctus a, eleifend nec, semper at, lorem. Sed pede. Nulla lorem metus, adipiscing ut, luctus sed, hendrerit vitae, mi.

</p>

</div>

</div>

<select name="effects" id="effectTypes">

<option value="blind">Blind</option>

<option value="bounce">Bounce</option>

<option value="clip">Clip</option>

<option value="drop">Drop</option>

<option value="explode">Explode</option>

<option value="fade">Fade</option>

<option value="fold">Fold</option>

<option value="highlight">Highlight</option>

<option value="puff">Puff</option>

<option value="pulsate">Pulsate</option>

<option value="scale">Scale</option>

<option value="shake">Shake</option>

<option value="size">Size</option>

<option value="slide">Slide</option>

<option value="transfer">Transfer</option>

</select>

<a href="#" id="button" class="ui-state-default ui-corner-all">Run Effect</a>

</body>

</html>

# JQuery function

## Arrays in JQuery

### Each()

<script src="jquery-1.10.2.js"></script>

<script>

// Метод each() - перебирает элементы jQuery-коллекции и запускает функцию для каждого элемента.

$(function () {

// i - порядковый номер элемента

$("p").each(function (i) {

// this - текущий элемент из коллекции

var txt = $(this).text();

$(this).text(i + 1 + " " + txt);

});

});

</script>

</head>

<body>

<p>Параграф</p>

<p>Параграф</p>

<p>Параграф</p>

<p>Параграф</p>

<p>Параграф</p>

<p>Параграф</p>

<p>Параграф</p>

<p>Параграф</p>

<p>Параграф</p>

<p>Параграф</p>

</body>

</html>

### Each(index,element)

<title>Метод each()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// function (index, element) - функция, которая будет вызвана для каждого элемента, где index - индекс текущего элемента, а element - текущий элемент. Можно использовать this.

$(function () {

$("div").each(function (index, element) {

$(element).css("background-color", "red");

if ($(element).is("#stop")) {

// return false - прекратить перебор элементов

return false;

}

});

});

</script>

<style>

div {

width: 30px;

height: 30px;

border: 1px solid red;

margin: 5px;

float:left;

}

</style>

</head>

<body>

<div></div>

<div></div>

<div></div>

<div></div>

<div></div>

<div id="stop">X</div>

<div></div>

<div></div>

<div></div>

<div></div>

</body>

### Grep()

<script>

// Метод grep(array, callback, invert) - находит элементы массива, которые удовлетваряют условию.

// array - массив в котором необходимо производить поиск

// callback - функция, которая выполняет проверку

// invert(true/false) - invert: если параметр "invert" указан как false, либо вообще не указан, функция вернет массив состоящий из всех элементов для которых функция "callback" вернула true. Если "invert" указана как true, функция вернёт массив состоящий из всех элементов для которых функция "callback" вернула false.

// Оригинальный массив остается прежним.

$(function () {

var arr = [1, 5, 9, 3, 8, 6, 1, 5, 9, 4, 7, 3, 8, 6, 9, 1];

$("div").text(arr.join(", "));

arr = $.grep(arr, function (element, index) {

// выбрать все элементы которые не равны 5 и индекс которых больше 4

return (element !== 5 && index > 4);

});

$("p").text(arr.join(", "));

arr = $.grep(arr, function (element) {

// все элементы не равные 9

return element !== 9;

});

$("span").text(arr.join(", "));

});

</script>

<style>

p {

color: Red;

}

</style>

</head>

<body>

<div></div>

<p></p>

<span></span>

</body>

</html>

Map()<title>Метод map()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// Метод map(array, function) - преобразование массива или объекта в новый массив.

// array - массив для преобразования

// function - функция для выполнения преобразования значений

// Оригинальный массив остается прежним.

$(function () {

var arr = ["Alex", "Max", "Corwin", "Daniel", "Den"];

$("div").text(arr.join(", "));

arr = $.map(arr, function (element, index) {

return (element.toUpperCase() + index);

});

$("p").text(arr.join(", "));

arr = $.map(arr, function (element) {

return element + element;

});

$("span").text(arr.join(", "));

});

</script>

<style>

p {

color: Red;

}

</style>

</head>

<body>

<div></div>

<p></p>

<span></span>

</body>

### InArray()

<title>Метод inArray()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// $.inArray(value, array) - возвращает индекс элемента, если он есть в массиве или -1, если ничего не найдено. value - то, что будем искать, array - имя массива, в котором будем искать.

$(function () {

var arr = ["a", "b", "c", "d", "e"];

var item = $.inArray("b", arr);

alert("Индекс найденного элемента - " + item);

});

</script>

<style>

p {

color: Red;

}

</style>

</head>

<body>

</body>

</html>

### MakeArray()

<title>Метод .makeArray()</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

//$.makeArray(obj) - преобразовывает массиво-подобный объект с выбранными элементами в обычный массив

var arr = $.makeArray($("p"));

arr.reverse();

$(arr).appendTo(document.body);

});

</script>

<style>

p {

color: Blue;

}

</style>

</head>

<body>

<p>Параграф №1</p>

<p>Параграф №2</p>

<p>Параграф №3</p>

<p>Параграф №4</p>

<p>Параграф №5</p>

<p>Параграф №6</p>

<p>Параграф №7</p>

<p>Параграф №8</p>

<p>Параграф №9</p>

<p>Параграф №10</p>

</body>

### Extend()

<title>Метод extend()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// extend - содержимое двух или большего количества объектов добавляет в объект, переданный первым параметром.

var target = { a: 1 };

var source1 = { b: 4, c: 5 };

var source2 = { d: 6, e: 7 };

$.extend(target, source1, source2); // target = {a: 1, b: 2, c: 4, d: 7, e: 6}

// проверить вывод в консоле браузера (не работает старых версиях IE)

console.dir(target);

console.log("Hello world");

</script>

<style>

p {

color: Blue;

}

div a {

margin-top: 10px;

margin-left: 30px;

}

</style>

</head>

<body>

</body>

</html>

### Extend() object

<title>Метод extend()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// $. - создание новых методов для глобального объекта jQuery

$.extend({

// Функция для поиска наименьшего

min: function (a, b)

{

return a < b ? a : b;

},

// Функция для поиска наибольшего

max: function (a, b) {

return a > b ? a : b;

},

// Функция для нахождения суммы

sum: function () {

var sum = 0;

for (var i = 0; i < arguments.length; i++) {

sum += arguments[i];

}

return sum;

}

});

var minV = $.min(2, 3);

var maxV = $.max(4, 5);

var sum = $.sum(4, 5, 8, 7, 5);

document.write("Минимальное значение: " + minV + "<br />");

document.write("Максимальное значение: " + maxV + "<br />");

document.write("Сумма: " + sum + "<br />");

</script>

</head>

<body>

</body>

### Extend() functions

<title>jQuery.fn.extend()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// $.fn - рассширение методов объектов, созданных с помощью jQuery

$(function () {

$.fn.extend({

check: function () {

return this.each(function () {

this.checked = true;

});

},

uncheck: function () {

return this.each(function () {

this.checked = false;

});

}

});

$("input[type='checkbox']").check();

});

</script>

<style>

label {

display: block;

margin: .5em;

}

</style>

</head>

<body>

<label><input type="checkbox" name="foo" /> Foo</label>

<label><input type="checkbox" name="bar" /> Bar</label>

</body>

### Index()

<title>Метод index()</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

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

// Метод index() - производит поиск среди выбранных элементов и возвращает индекс найденого элемента. Отсчет начинается с нуля.

var index = $("div").index(this);

$("span").text("That was div index #" + index);

});

});

</script>

<style>

div {

background: yellow;

margin: 5px;

}

span {

color: red;

}

</style>

</head>

<body>

<span>Click a div!</span>

<div>First div</div>

<div>Second div</div>

<div>Third div</div>

</body>

</html>

## Data

### Data()

<title>Метод data()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// data - добавляет или считывает дополнительные данные элемента.

$(function () {

// добавление к каждому параграфу доп. данных по ключу name

$("p").eq(0).data("name", "first paragraph");

$("p").eq(1).data("name", "second paragraph");

$("p").eq(2).data("name", "thrid paragraph");

$("p").eq(3).data("name", "fourth paragraph");

$("p").eq(4).data("name", "fifth paragraph");

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

// чтение данных по ключу name из кликнутого параграфа

var d = $(this).data("name");

$("span").text(d);

});

});

</script>

</head>

<body>

<p>Параграф №1</p>

<p>Параграф №2</p>

<p>Параграф №3</p>

<p>Параграф №4</p>

<p>Параграф №5</p>

<span></span>

</body>

</html>

### Data() element object

<title>Метод data()</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

$("p:first").data("obj",

{

first: 123,

second: "Text",

});

var dataP = $("p:first").data("obj");

alert(dataP.first + " " + dataP.second);

});

</script>

</head>

<body>

<p>Параграф №1</p>

</body>

</html>

### RemoveData()

<title>Метод removeData()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// .removeData() - позволяет удалить все данные элемента, которые были добавлены через метод data.

// .removeData(key) - позволяет удалить данные по заданному ключу.

$(function () {

$("p").data("name1", "abc");

$("p").data("name2", "123");

var dataP = $("p").data();

alert(dataP.name1 + " " + dataP.name2); // abc 123

$("p:first").removeData("name2");

alert(dataP.name1 + " " + dataP.name2); // abc undefined

});

</script>

</head>

<body>

<p>Параграф №1</p>

</body>

</html>

### RemoveData() example

<title>Метод removeData()</title>

<script src="jquery-1.10.2.js"></script>

<script>

$(function () {

$("p").data("name1", "value 1");

$("p").data("name2", "value 2");

$("p").data("name3", "value 3");

$("p").data("name4", "value 4");

$("p").data("name5", "value 5");

$("p").data("name6", "value 6");

var dataP = $("p:first").data();

for (item in dataP) {

$("span").eq(0).html($("span").html() + item + " " + dataP[item] + "<br/>");

}

$("p").removeData("name3 name4 name5");

for (item in dataP) {

$("span").eq(1).html($("span").eq(1).html() + item + " " + dataP[item] + "<br/>");

}

});

</script>

</head>

<body>

<p>Параграф №1</p>

<span></span>

<br />

<br />

<span></span>

</body>

</html>

## NoCondflict

### NoConflict()

<title>Метод noConflict()</title>

<script src="jquery-1.10.2.js"></script>

<script>

// jQuery.noConflict() - отказываемся от использования переменной $ как объекта-контейнера для функциональности jQuery.

// данный подход часто используется для того, чтобы решить конфликты при исползовании других JavaScript библиотек вместе с jQuery.

$(function () {

var newJ = jQuery.noConflict();

newJ("#div1").css("background-color", "green");

jQuery("#div2").css("background-color", "green"); // Глобальная переменная jQuery продолжает работать

$("#div3").css("background-color", "green"); // Глобальная переменная $ перестает использоваться полсе вызова noConflict

});

</script>

<style>

div {

width: 200px;

height: 200px;

background-color: red;

margin: 20px;

}

</style>

</head>

<body>

<div id="div1"></div>

<div id="div2"></div>

<div id="div3"></div>

</body>

</html>

### NoConflict(true)

<script>

$(function () {

var newJ = jQuery.noConflict(true); // true - удалить все глобальные переменные, которые использует jQuery.

newJ("#div1").css("background-color", "Green");

jQuery("#div2").css("background-color", "Green");

$("#div3").css("background-color", "Green");

});

</script>

# Ajax

## XMLHttpRequest

### Sync example

<title>XMLHttpRequest синхронный запрос</title>

<script>

window.onload = function () {

document.getElementById("btnGet").onclick = function () {

var xhr = new XMLHttpRequest(); // Создание объекта для HTTP запроса.

xhr.open("GET", "testfile.html", false); // Настройка объекта для отправки синхронного GET запроса

xhr.send(); // Отправка запроса, так как запрос является синхронным, следующая строка кода выполнится только после получения ответа со стороны сервера.

document.getElementById("output").innerHTML += xhr.responseText; // responseText - текст ответа полученного с сервера.

}

}

</script>

</head>

<body>

<input id="btnGet" type="button" value="Get Info" />

<p id="output"></p>

</body>

</html>

File testfile.html

<b>Hello world</b> <br />

### Async example

<title>XMLHttpRequest асинхронный запрос</title>

<script>

window.onload = function () {

document.getElementById("btnGet").onclick = function () {

var xhr = new XMLHttpRequest(); // Создание объекта для HTTP запроса.

xhr.open("GET", "testfile.html", true); // Настройка объекта для отправки асинхронного GET запроса

// функция-обработчик срабатывает при изменении свойства readyState

// Значения свойства readyState:

// 0 - Метод open() еще не вызывался

// 1 - Метод open() уже был вызван, но метод send() еще не вызывался.

// 2 - Метод send() был вызван, но ответ от сервера еще не получен

// 3 - Идет прием данных от сервера. Для значения 3 Firefox вызывает обработчик события несколько раз IE только один раз.

// 4 - Ответ от сервера полностью получен (Запрос успешно завершен).

xhr.onreadystatechange = function () {

if (xhr.readyState == 4) { // если получен ответ

if (xhr.status == 200) { // и если статус код ответа 200

document.getElementById("output").innerHTML += xhr.responseText; // responseText - текст ответа полученного с сервера.

}

}

}

xhr.send(); // Отправка запроса, так как запрос асинхронный сценарий продолжит свое выполнение. Когда с сервера придет ответ сработает событие onreadystatechange

}

}

</script>

</head>

<body>

<input id="btnGet" type="button" value="Get Info" />

<p id="output"></p>

</body>

</html>

## JQuery ajax

### AjaxSetup

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

<script type="text/javascript">

// обработчик события загрузки страницы.

window.onload = function () {

// Создание объекта запроса.

// $.ajaxSetup() - установка глобальных настроек для запросов $.ajax.

$.ajaxSetup({

// адрес для запроса

url: "test.html",

// тип запроса, например, GET, POST, PUT

type: "GET",

});

// $.ajax() возвращает объект XMLHttpRequest. В большинстве случаев Вам не понадобится работать непосредственно с этим объектом, но он все же

// доступен, в случае, если Вам необходимо прервать запрос вручную.

// отправка запроса на сервер, при этом используются параметры установленные методом $.ajaxSetup

$.ajax({

// функция, которая запускается в результате успешной обработки запроса.

success: function (msg) {

alert("Данные получены от сервера: " + msg);

}

});

}

</script>

</head>

<body>

<p></p>

</body>

</html>

### Ajax datatype

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

<script type="text/javascript">

window.onload = function () {

// Создание объекта запроса.

// $.ajax() возвращает объект XMLHttpRequest.

$.ajax({

type: "GET",

url: "test.js",

// определение типа ответа, который ожидается со стороны сервера

dataType: "script" // json, text, xml, script

});

}

</script>

</head>

<body>

<p></p>

</body>

</html>

Test.js

alert("Hello Ajax!");

$("p").text("Hello Ajax!!!");

### Ajax success

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

<script type="text/javascript">

window.onload = function () {

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

$.ajax({

type: "GET",

url: "test2.html",

success: function (data) {

$("p:first").append(data);

}

});

});

}

</script>

</head>

<body>

<button id="click">Запрос</button>

<p></p>

</body>

</html>

Test2.html

<p>Hello world</p>

### Ajax data

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

<script type="text/javascript">

// обработчик события загрузки страницы.

window.onload = function () {

$("#loader").hide();

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

$("#loader").show();

var dataToSend = "";

dataToSend += "login=" + $("#login").val();

dataToSend += "&pass=" + $("#password").val();

$.ajax({

type: "POST",

url: "HandlerData.ashx",

// данные, которые будут отправлятся на сервер с запросом

data: dataToSend,

success: function (data) {

$("#output").text(data);

$("#loader").hide();

}

});

});

}

</script>

</head>

<body>

Login <input id="login" type="text" />

Password <input id="password" type="text" />

<br />

<button>Войти</button> <span id="loader"> Загрузка</span>

<p id="output"></p>

</body>

</html>

HandlerData.ashx

namespace AJAX\_Basics\_JavaScript

{

public class Handler : IHttpHandler

{

public void ProcessRequest(HttpContext context)

{

Thread.Sleep(2000);

var parametrs = context.Request.Form;

context.Response.ContentType = "text/plain";

// проверка параметров

if (parametrs["login"] == "user1" && parametrs["pass"] == "12345")

{

// отправка ответа

context.Response.Write("Авторизация прошла успешно");

}

else

{

context.Response.Write("Ошибка авторизации");

}

}

public bool IsReusable

{

get

{

return true;

}

}

}

}

### Ajax xml

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

<script>

window.onload = function () {

function send() {

var xhr;

if (window.XMLHttpRequest) {// code for IE7+, Firefox, Chrome, Opera, Safari

xhr = new XMLHttpRequest();

}

else {// code for IE6, IE5

xhr = new ActiveXObject("Microsoft.XMLHTTP");

}

xhr.open("GET", "note.xml", false);

xhr.send(); // отправка запроса и ожидание ответа

return xhr.responseXML;

}

function print(xmlDoc) {

var div1 = document.getElementById("to"),

div2 = document.getElementById("from"),

div3 = document.getElementById("message"),

toElement = xmlDoc.getElementsByTagName("to"),

fromElement = xmlDoc.getElementsByTagName("from"),

messageElement = xmlDoc.getElementsByTagName("body");

div1.innerHTML += toElement[0].childNodes[0].nodeValue;

div2.innerHTML += fromElement[0].childNodes[0].nodeValue;

div3.innerHTML += messageElement[0].childNodes[0].nodeValue;

}

document.getElementById("sendRequest").onclick = function () {

var response = send();

print(response);

}

}

</script>

</head>

<body>

<div id="to"><b>To:</b> </div>

<div id="from"><b>From:</b> </div>

<div id="message"><b>Message:</b> </div>

<button id="sendRequest">Запрос</button>

</body>

</html>

Note.xml

<note>

<to>user@example.com</to>

<from>sender@example.com</from>

<heading>New Message</heading>

<body>Hello! How are you?</body>

</note>

### Ajax(xml)

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

<script>

$(document).ready(function () {

$("#sendRequest").click(function () {

$.ajax({

url: "note.xml",

type: "GET",

dataType: "xml",

success: function (xml) {

$("#to").append($(xml).find("to").text());

$("#from").append($(xml).find("from").text());

$("#message").append($(xml).find("body").text());

},

// в случае ошибки

error: function (xhr, status, error) {

// staus - возможные значения "timeout", "error", "abort", and "parsererror"

alert("ERROR: " + xhr.status + " " + error);

}

});

});

});

</script>

</head>

<body>

<div id="to"><b>To:</b> </div>

<div id="from"><b>From:</b> </div>

<div id="message"><b>Message:</b> </div>

<button id="sendRequest">Запрос</button>

</body>

</html>

### Load()

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title></title>

<script src="jquery-1.8.1.js" type="text/javascript"></script>

<script type="text/javascript">

$(document).ready(function () {

// .load() - загружает HTML из удаленного файла и встраивает его в объектную модель документа (DOM).

$("#objectID").load("test.html #content", // адрес файла и селектор для получения данных

function () {

alert("Loaded!");

});

});

</script>

</head>

<body>

<div id="objectID"></div>

</body>

</html>

Test.html

<p>

Hello world

</p>

<div id="content">

Test content

</div>

## Ajax Events

### Events

<script type="text/javascript" src="jquery-1.8.1.js"></script>

<script type="text/javascript">

$(document).ready(function () {

$("#action").click(function send() {

$.ajax({

type: "GET",

url: "test.html",

// beforeSend - событие происходит до отправки запроса на сервер.

// можно изменить настройки объекта XMLHttpRequest (например, установить дополнительные заголовки)

beforeSend: function (xhr) {

$("#msg").append("<li>beforeSend (local)</li>");

},

// success - событие происходит только в результате успешного завершения обработки запроса (если с сервера не вернулись ошибки)

success: function (data) {

$("#msg").append("<li>sucess (local)</li>");

},

// error - срабатывает если сервер вернул ошибку

error: function (data) {

$("#msg").append("<li>error (local)</li>");

},

// complete - срабатывает при завершении запроса не важно усрешно он был обработан или нет

complete: function () {

$("#msg").append("<li>complete (local)</li>");

}

});

});

// ajaxStart (глобальное событие) - это событие запускаеться если стартует AJAX запрос и в данный момент нет других AJAX запросов

$(document).ajaxStart(function (event) {

$("#msg").append("<li>ajaxStart (global).</li>");

});

// ajaxStart (глобальное событие) - происходит перед отправкой запроса

$(document).ajaxSend(function (event) {

$("#msg").append("<li>ajaxSend (global).</li>");

});

// ajaxSuccess (глобальное событие) - происходит только если запрос завершился успешно

$(document).ajaxSuccess(function (event) {

$("#msg").append("<li>ajaxSuccess (global).</li>");

});

// ajaxError (глобальное событие) - срабатывает если сервер вернул ошибку

$(document).ajaxError(function (event) {

$("#msg").append("<li>ajaxError (global).</li>");

});

// ajaxComplete (глобальное событие) - срабатывает при завершении запроса не важно усрешно он был обработан или нет

$(document).ajaxComplete(function (event) {

$("#msg").append("<li>ajaxComplete (global).</li>");

});

// ajaxStop (глобальное событие) - больше не осталось асинхронных запросов

$(document).ajaxStop(function (event, request, settings) {

$(this).append("<li>ajaxStop (global).</li>");

});

});

</script>

</head>

<body>

<button id="action">Send Request</button>

<ul id="msg"></ul>

</body>

</html>

## Object to url parameters

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

<script type="text/javascript">

$(document).ready(function () {

// param() - Преобразует объект, массив или массив объектов в строку, которая соответствует формату передачи данных через url.

var obj = {

one: 1,

two: 2,

};

var urlParams = $.param(obj);

alert(urlParams);

var obj2 = {

a: {

one: 1,

two: 2,

},

b: [1, 2, 3]

};

var urlParam2 = $.param(obj2);

alert(urlParam2);

var decoded = decodeURIComponent(urlParam2);

alert(decoded);

});

</script>

</head>

<body>

</body>

</html>

## Serialize

<head>

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

</head>

<body>

<form>

<input type="text" name="login" value="abc" />

<input type="text" name="password" value="123" />

</form>

<script>

// serialize() - Возвращает строку, с именами и значениями выбранных элементов формы. Эта строка будет иметь формат параметров url-запроса.

var data = $('form').serialize();

alert(data);

</script>

</body>

</html>

## SerializeArray()

<head>

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

</head>

<body>

<form>

<input type="text" name="login" value="abc" />

<input type="text" name="password" value="123" />

</form>

<script type="text/javascript">

//serializeArray() - возвращает массив объектов, содержащий данные элементов формы.

var arr = $('form').serializeArray();

alert(arr[0].value + " " + arr[1].value);

</script>

</body>

</html>

## JSON

### Json serialize/desirialize

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

<script type="text/javascript">

// Создаем объект JavaScript

var student = {

name: "Ivan",

age: 25,

course: "jQuery"

};

// JSON.stringify() - метод для сериализации объекта в JSON

var data = JSON.stringify(student); // сериализация

alert(data); // '{ "name" : "Ivan", "age": "25", "course" : "jQuery" }'

// JSON.parse() - метод для десериализации данных из JSON в JavaScript объект

var obj = JSON.parse(data);

alert(obj.name + "\r\n" + obj.age + "\r\n" + obj.course);

</script>

</head>

<body>

</body>

</html>

### Json server handler

<title></title>

<script type="text/javascript" src="jquery-1.8.1.js"></script>

<script type="text/javascript">

var json;

// обработчик события загрузки страницы.

window.onload = function () {

// Создание объекта запроса.

$.ajax({

type: "GET",

url: "HandlerObject.ashx",

dataType: "json",

// функция получает десериализованный объект

success: function (obj) {

$(document.body).append("<p>" + obj.Name + "</p>");

$(document.body).append("<p>" + obj.Age + "</p>");

$(document.body).append("<p>" + obj.Course + "</p>");

}

});

}

</script>

</head>

<body>

</body>

</html>

HandlerObject.ashx

namespace AJAX\_Basics\_JavaScript

{

/// <summary>

/// Summary description for Handler1

/// </summary>

public class Handler1 : IHttpHandler

{

class Student

{

public string Name { get; set; }

public int Age { get; set; }

public string Course { get; set; }

}

public void ProcessRequest(HttpContext context)

{

context.Response.ContentType = "json/application";

JavaScriptSerializer js = new JavaScriptSerializer();

Student st = new Student();

st.Name = "Ivan";

st.Age = 25;

st.Course = "jQuery";

context.Response.Write(js.Serialize(st));

}

public bool IsReusable

{

get

{

return false;

}

}

}

}