

— JAVASCRIPT —

jQuery. jQuery UI

jQuery

динамическая функциональность страниц

находить один или несколько элементов страницы, а затем выполняете над ними некоторые действия

Функция `$()`

находить элементы на странице

добавлять "на лету" новый html

```
// создадим div-элемент и добавим его в конец элемента с идентификатором content  
$("<div><p>Ба-бах!</p></div>").appendTo("#content");
```

привязать всю функциональность jQuery к уже существующим объектам DOM-элементов

```
// найдем элемент с идентификатором some_id, средствами обычного javascript  
var el = document.getElementById("some_id");  
  
// установим у этого элемента новое значение css-свойства margin  
$(el).css("margin", "5px");
```

jQuery

окончанию загрузки всей страницы

```
window.onload = function(){  
    // вызов нужных функций скрипта  
}
```

момент готовности дерева DOM

```
1 | $( document ).ready(function() {  
2 |     // Handler for .ready() called.  
3 | });
```

```
1 | $(function() {  
2 |     // Handler for .ready() called.  
3 | });
```

jQuery

ПОИСК ЭЛЕМЕНТОВ

<code>\$("div")</code>	вернет все div-элементы на странице.
<code>\$(".someBlock")</code>	вернет все элементы с классом someBlock.
<code>\$("#content")</code>	вернет элемент с идентификатором content.
<code>\$("#content2 div.someBlock")</code>	вернет div-элементы с классом someBlock, которые находятся внутри элемента с идентификатором content2.
<code>\$("div:odd")</code>	вернет div-элементы, находящиеся на странице под нечетными номерами.
<code>\$("[value = 5]")</code>	вернет все элементы с атрибутом value, равным 5.

jQuery

Селекторы по атрибутам

"[name]"	элементы, содержащие атрибут <i>name</i>
"[name = value]"	элементы, у которых значение атрибута <i>name</i> совпадает с <i>value</i>
"[name != value]"	элементы, у которых значение атрибута <i>name</i> не совпадает с <i>value</i>
"[name ^= value]"	элементы, у которых значение атрибута <i>name</i> начинается с <i>value</i>
"[name \$= value]"	элементы, у которых значение атрибута <i>name</i> заканчивается на <i>value</i>
"[name *= value]"	элементы, у которых значение атрибута <i>name</i> содержит подстроку <i>value</i>

Фильтры по содержимому

":contains(text)"	элементы, содержащие заданный текст
":empty"	элементы без содержимого (без текста и других элементов)
":has(selector)"	элементы, которые содержат хотя бы один элемент из <i>selector</i>
":parent"	непустые элементы

Фильтры дочерних элементов

Данные селекторы отфильтровывают элементы по их расположению в родительских элементах

":first-child"	элементы, расположенные первыми в своих родительских элементах
":last-child"	элементы, расположенные последними в своих родительских элементах
":nth-child()"	элементы, расположенные определенным образом в родительских элементах (четные, нечетные, идущие под заданным номером)
":nth-child-last()"	
":only-child"	элементы, являющиеся единственными потомками в своих родительских элементах

Простые фильтры

Фильтры, с помощью которых можно уточнить результат других селекторов. По назначению и синтаксису они очень похожи на псевдоклассы в CSS

":focus"	элемент, находящийся в фокусе
":first"	первый найденный элемент
":last"	последний найденный элемент
":eq()"	элемент идущий под заданным номером среди выбранных
":not(selector)"	все найденные элементы, кроме указанных в <i>selector</i>
":even"	элементы с четными номерами позиций, в наборе выбранных элементов
":odd"	элементы с нечетными номерами позиций, в наборе выбранных элементов

jQuery

манипуляции и эффекты

<code>\$("#bigIt").css("height")</code>	возвратит значение высоты у элемента с идентификатором bigIt.
<code>\$("#div").css("width", "20px")</code>	установит новое значение ширины всем div-элементу на странице.
<code>\$("#bigIt").attr("class")</code>	возвратит значение класса элемента с id = bigIt.
<code>\$("#bigIt").attr("class", "box")</code>	установит новое значение атрибута class у элемента с id = bigIt.
<code>\$("#bigIt").html(<p>Новье!</p>)</code>	изменит все html-содержимое элемента с id = bigIt, на заданное в методе html.
<code>\$("#bigIt").text()</code>	возвратит текст, находящийся внутри элемента с id = bigIt.
<code>\$(".someBox").empty()</code>	очистит от содержимого элементы с классом someBox.

```
$("#bigIt").empty().attr("class", "noContent");  
// в результате, у элемента с идентификатором bigIt будет удалено все содержимое,  
// после чего ему будет установлен класс noContent.
```

```
$("#div")           // найдем все div-элементы  
.parent()           // найдем их родительские элементы  
.css("height", "10px") // установим последним высоту в 10 пикселей  
.fadeTo(0, 0.5)      // установим им (родителям div'ов) прозрачность в 50%  
.addClass("divOwner"); // добавим им же класс divOwner
```

jQuery

Работа с набором элементов

<code>\$("div").parent()</code>	вернет родительские элементы всех div-ов.
<code>\$("div").children()</code>	вернет дочерние элементы всех div-ов.
<code>\$("#someId").next()</code>	вернет элемент, лежащий сразу после someId.
<code>\$("div").prev()</code>	вернет элементы, лежащие перед div'ами.
<code>\$("div").eq(i)</code>	вернет div-элемент, с индексом i в наборе.
<code>\$("div").get(i)</code>	вернет DOM-объект div'a, с индексом i.
<code>\$("div").get()</code>	вернет массив DOM-объектов всех div-ов.
<code>\$("div").size()</code>	вернет размер набора (количество div-ов).

jQuery

Ajax

<code>\$.get()</code>	производит запрос к серверу методом GET
<code>\$.post()</code>	производит запрос к серверу методом POST
<code>.load()</code>	производит запрос HTML-данных у сервера и помещает их в выбранные элементы страницы
<code>\$.getJSON()</code>	производит запрос JSON-данных у сервера методом GET
<code>\$.getScript()</code>	производит запрос файла javascript методом GET, а затем выполняет код из полученного файла.

```
$.ajax({  
  url: 'ajax/test.html',  
  success: function(){  
    alert('Load was performed.');  }  
});
```

beforeSend происходит непосредственно перед отправкой запроса на сервер.

error происходит в случае неудачного выполнения запроса.

dataFilter происходит в момент прибытия данных с сервера. Позволяет обработать "сырые" данные, присланные сервером.

success происходит в случае удачного завершения запроса.

complete происходит в случае любого завершения запроса.

jQuery

События

<u>.on()</u>	Универсальный метод для установки обработчиков событий на выбранные элементы страницы.
<u>.off()</u>	Удаляет обработчики, установленные с помощью <code>.on()</code> .

<u>.click()</u>	Устанавливает обработчик "клика" мышью по элементу, либо, запускает это событие.
<u>.dblclick()</u>	Устанавливает обработчик двойного "клика" мышью по элементу, либо, запускает это событие.
<u>.hover()</u>	Устанавливает обработчик двух событий: появления/исчезновения курсора над элементом.
<u>.mousedown()</u>	Устанавливает обработчик нажатия кнопки мыши, либо, запускает это событие.
<u>.mouseup()</u>	Устанавливает обработчик поднятия кнопки мыши, либо, запускает это событие.
<u>.mouseenter()</u>	Устанавливает обработчик появления курсора в области элемента, либо, запускает это событие. Появление этого события, отработано лучше, чем стандартного <code>mouseover</code> .
<u>.mouseleave()</u>	Устанавливает обработчик выхода курсора из области элемента, либо, запускает это событие. Появление этого события, отработано лучше, чем стандартного <code>mouseout</code> .
<u>.mousemove()</u>	Устанавливает обработчик движения курсора в области элемента, либо, запускает это событие.
<u>.mouseout()</u>	Устанавливает обработчик выхода курсора из области элемента, либо, запускает это событие.
<u>.mouseover()</u>	Устанавливает обработчик появления курсора в области элемента, либо, запускает это событие.

<u>.keydown()</u>	Устанавливает обработчик перехода клавиши клавиатуры в нажатое состояние, либо, запускает это событие.
<u>.keyup()</u>	Устанавливает обработчик возвращение клавиши клавиатуры в ненажатое состояние, либо, запускает это событие.
<u>.keypress()</u>	Устанавливает обработчик ввода символа с клавиатуры, либо, запускает это событие.

jQuery

```
// Установим обработчик нажатия кнопкой мыши, элементу с идентификатором foo  
$('#foo').on('click', function(){  
    alert('Вы нажали на элемент "foo"');  
});
```

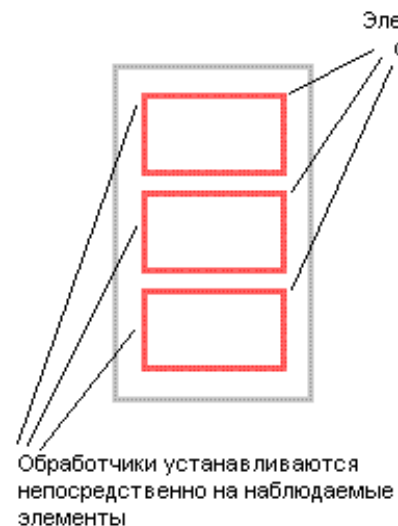
// Теперь, при нажатии на элемент foo, будет выведено сообщение

```
// неэффективное выполнение задачи - элемент с id=elm  
// ищется заново при каждом выполнении обработчика  
$('.mousemove-elm').on('mousemove', function(event){  
    $("#elm").text(event.pageX + ", " + event.pageY);  
});
```

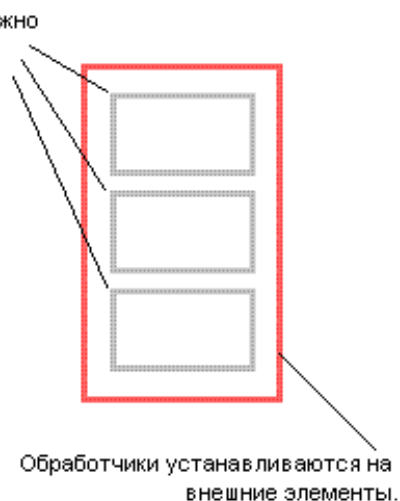
```
// эффективное выполнение задачи - элемент с id=elm  
// ищется только один раз  
var elm = $("#elm");  
$('.mousemove-elm').on('mousemove', function(event){  
    elm.text(event.pageX + ", " + event.pageY);  
});
```

jQuery

Непосредственная обработка событий



Делегированная обработка событий



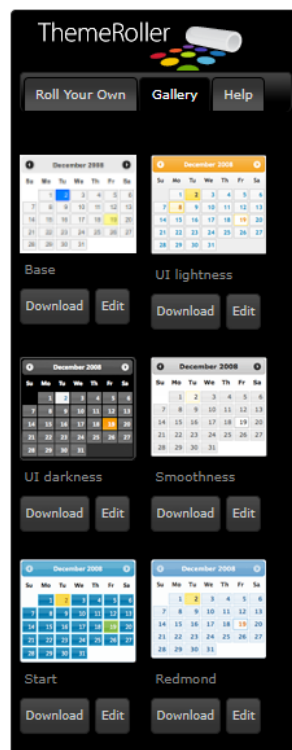
```
$('#ul').on('click', 'li', function(){ // обработчик ....
```

```
$('#ul li').on('click', function(){ // обработчик ....
```

— jQuery UI —

jQuery UI — это библиотека на основе jQuery, реализующая более 20 плагинов, среди которых плагины организующие различное поведение (например перетаскивание или растягивание элементов), восемь видов виджетов (такие как календарь, диалоговые окна, систему вкладок и.т.д) и анимационные эффекты.

ThemeRoller



Accordion

▼ First

Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet.

▸ Second

▸ Third

Autocomplete

Button

A button element



Checkboxradio

☒ Choice 1

☐ Choice 2

☐ Choice 3

— jQuery UI —

```
<link type="text/css" href="css/themename/jquery-ui-1.8.12.custom.css" rel="Stylesheet" />  
<script type="text/javascript" src="js/jquery-1.4.4.min.js"></script>  
<script type="text/javascript" src="js/jquery-ui-1.8.12.custom.min.js"></script>
```

— jQuery UI —

Вызов методов у ui элементов

```
$("#someId").pluginName("имя метода", параметры метода);
```

<code>\$("#someId").dialog("close")</code>	метод <code>close</code> закрывает диалоговое окно, установленное на элемент с <code>id = someId</code>
<code>\$("div").draggable("destroy")</code>	<code>destroy</code> удалит функциональность <code>draggable</code> со всех <code>div</code> -элементов на странице

— jQuery UI —

Свойства ui элементов

```
// сделаем из элемента с id=someId диалоговое окно с помощью  
// плагина dialog и укажем при этом заголовок для окна  
$("#someId").dialog({title:"Сообщение"});  
  
// сделаем из первого div'a на странице календарь с помощью  
// плагина datePicker, и укажем минимальную и максимальную дату  
$("#div:first").datepicker({  
    minDate:new Date(2007, 1 - 1, 1),  
    maxDate:new Date(2013, 1 - 1, 1)  
});
```

— jQuery UI —

События ui элементов

```
// обработка события close диалогового окна  
$("selector").dialog({  
    close: function(event, ui) { ... }  
});
```


— jQuery UI —

```
$("#draggable").draggable(); // эта строка кода, которая делает элемент перетаскиваемым
```

```
$("#el").draggable({ drag:function(event, ui){  
    ui.position.left = x;  
    ui.position.top = y;  
}});
```

```
$(this).data('draggable').offset.click.top = y;
```

```
▶ .draggable("destroy")
```

```
▶ .draggable("disable")
```

```
▶ .draggable("enable")
```

```
▼ .draggable("option", optionName, [value])
```

↓ Внутреннее имя события : внешнее имя

```
▶ create:dragcreate
```

```
▶ start:dragstart
```

```
▶ drag:drag
```

```
▶ stop:dragstop
```

— JAVASCRIPT —

jQuery. jQuery UI