

## Базовые фильтры

**:first** - выбирает первый элемент соответствующего селектора.

```
$("label:first").css("font-style", "italic");
```

Данная инструкция выделит курсивом первый элемент label

**:last** - выбирает последний элемент соответствующего селектора

```
Пример: $("label:last").css("font-style", "italic");
```

выделит курсивом последний элемент label

**:even** - выбирает четные элементы (отсчет от нуля).

```
$("tr:even").css("font-weight", "bold")
```

Выделит жирным шрифтом четные строки

**:odd** - выбирает нечетные элементы (отсчет от нуля).

```
$("tr:odd").css("font-style", "italic"); -
```

курсивом все нечетные строки таблицы (т.к. нумерация идет с нуля, то зрительно кажется, что четные строки).

**:eq(index)** - выбирает элемент по его индексу (начиная с нуля).

**\$("#td:eq(2)").css("color", "red")** Выделит красным цветом текст второй ячейки таблицы (т.к. нумерация идет с нуля, то зрительно кажется, что третьей ячейки).

**:gt(index)** - выбирает элементы с индексом больше указанного.

Пример: **\$("#td:gt(2)").css("color", "blue");** Выделит красным цветом текст во всех ячейках таблицы, начиная со 2

**:lt(index)** - выбирает элементы с индексом меньше указанного.

**\$("#td:lt(4)").css("color", "green");** Выделит зеленым т курсивом текст в первых пяти ячейках таблицы

**:header** - выбирает все элементы, которые являются заголовками (h1, h2...)

**\$("#:header").css("color", "red");**

Данная инструкция сделает все заголовки красными

## Фильтры содержимого

**:contains (text)** - выбирает элементы, которые содержат заданный текст (text).

`$("#label:contains('Cube')").css("text-decoration", "underline");` - подчеркнет все элементы label, содержащие подстроку 'Cube'.

**:empty** - выбирает все элементы, которые не содержат потомков (т.е. являются пустыми).

Пример: `$("#td:empty").text("-----");`

Данная инструкция найдет все пустые ячейки таблицы и вставит в них текст "-----".

**:has(selector)** - выбирает элементы, которые содержат хотя бы один элемент, указанный в селекторе.

Пример: `$("#li:has(div)").css("border", "dotted 2px green");` Данная инструкция найдет те li-, в которых есть хотя бы один div и обведет рамкой

**:parent** - выбирает родительские элементы, т.е. те, у которых есть потомки.

Пример: `$("#td:parent").css("font-style", "italic");` Данная инструкция найдет все ячейки таблицы, в которых есть текст или еще что-либо, и сделает их шрифт курсивом.

## Перемещение по дереву в пределах одного иерархического уровня

### `next ()`, `next (селектор)`

Выбирает сестринские элементы, непосредственно следующие за каждым из элементов, содержащихся в объекте jQuery. Имеется дополнительная возможность фильтрации результатов с использованием селектора

```
<ul>
```

```
<li>list item 1</li>
```

```
<li>list item 2</li>
```

```
<li class="third-item">list item 3</li>
```

```
<li>list item 4</li>
```

```
<li>list item 5</li>
```

```
</ul>
```

```
$( "li.third-item" ).next().css( "background-color", "red" );
```

Результатом будет применение стиля css (фон – красный) к элементу выделенному на слайде красным цветом

## **nextAll() nextAll(селектор)**

Выбирает все последующие сестринские элементы для каждого из элементов, содержащихся в объекте jQuery. Имеется дополнительная возможность фильтрации результатов с использованием селектора

## **nextUntil(селектор),**

## **nextUntil(селектор, селектор)**

## **nextUntil(jQuery),**

## **nextUntil(jQuery, селектор),**

## **nextUntil(HTMLElement[]) nextUntil(HTMLElement[], селектор)**

Выбирает для каждого элемента последующие сестринские элементы вплоть до элемента (но не включая его), соответствующего селектору или содержащегося в объекте jQuery ИЛИ массиве HTMLElement []. Имеется дополнительная возможность фильтрации результатов с использованием селектора

## **prev(), prev (селектор)**

Выбирает сестринские элементы, непосредственно предшествующие каждому из элементов, содержащихся в объекте jQuery. Имеется дополнительная возможность фильтрации результатов с использованием селектора

## **prevAll(), prevAll(селектор)**

Выбирает все предшествующие сестринские элементы для каждого из элементов, содержащихся в объекте jQuery. Имеется дополнительная возможность фильтрации результатов с использованием селектора, передаваемого методу в качестве второго аргумента

## **prevUntil(селектор),**

## **prevUntil(селектор, селектор),**

## **prevUntil(jQuery), prevUntil(jQuery, селектор) prevUntil(HTMLElement[]) prevUntil(HTMLElement[], селектор)**

Выбирает для каждого элемента предшествующие сестринские элементы вплоть до элемента (но не включая его), соответствующего селектору или содержащегося в объекте jQuery или массиве HTMLElement []. Имеется дополнительная возможность фильтрации результатов с использованием селектора, передаваемого методу в качестве второго аргумента

**siblings(), siblings(селектор)** Выбирает все сестринские элементы для каждого из элементов, содержащихся в объекте jQuery. Имеется дополнительная возможность фильтрации результатов с использованием селектора

## Выбор всех сестринских элементов

### Метод `siblings()`

обеспечивает возможность выбора всех сестринских элементов для всех элементов, содержащихся в объекте jQuery.

Пример использования этого метода

```
$(document).ready(function() {  
    $('img[src*=Cube]').parent().siblings().css("border", "thick solid blue");  
});
```

В этом примере сначала выбираем 2 элемента `img`, затем находим их родительские элементы с помощью метода `parent()`, после чего выбираем сестринские элементы последних с помощью метода `siblings()`. При этом выбираются как предшествующие, так и последующие сестринские элементы и для свойства `border` каждого из них устанавливается определенное значение с помощью метода `css()`.

**Обратите внимание: выбираются лишь сестринские элементы, но не сами элементы.**

Эта ситуация изменится, если один из элементов, содержащихся в объекте jQuery, является сестринским по отношению к другому, как показано в коде

```
$('#row1 li.dcell').siblings().css("border", "thick solid blue");
```

В этом сценарии мы начинаем с выбора всех элементов li, являющихся дочерними по отношению к элементу row1, а затем вызываем метод siblings (). Каждый из элементов в выбранном наборе является сестринским в отношении по крайней мере одного из других элементов.

### **Выбор последующих и предшествующих сестринских элементов**

Пример использования методов nextAll () и prevAll ()

```
$(document).ready(function() {  
    $('img[src*=Duplicator]').parent().nextAll().css("border", "thick solid blue");  
    $('img[src*=FELIX]').parent().prevAll().css("border", "thick double red");  
});
```

В этом сценарии для родительского элемента изображения Duplicator выбираются все последующие сестринские элементы, а для родительского элемента изображения FELIX - все предыдущие



# Работа с содержимым элементов

## Методы для работы с содержимым элементов

**text()** -Получает объединенное текстовое содержимое всех элементов, содержащихся в объекте jQuery, и их потомков

**text(значение)** -Устанавливает текстовое содержимое для всех элементов, содержащихся в Объекте jQuery

**html()** -Возвращает HTML-содержимое первого из элементов, содержащихся в объекте jQuery

**html(значение)** -Устанавливает HTML-содержимое для всех элементов, содержащихся в объекте jQuery

**text(функция), html(функция)**

Устанавливает текстовое и HTML-содержимое с помощью функции

С методом text () связана одна особенность: если он вызывается без аргументов, то возвращаемый им результат генерируется на основе содержимого всех выбранных элементов. В отличие от этого поведение метода html () согласуется с поведением остальных методов jQuery, и он возвращает лишь содержимое первого элемента выбранного набора

```
$(document).ready(function(){  
var htmlContent = $('li.dcell').html();  
console.log(htmlContent);});
```

В этом сценарии метод `html ()` используется для считывания HTML-содержимого первого из элементов, соответствующих селектору `li.dcell`, и вывода на консоль результата, который представлен ниже.

```
  
  <div class="bot_part">  
    <label for="pr_1">3D принтер Duplicator 4x:</label> <input  
name="pr_1" type="text" value="0" required="">  
  </div>
```

Обратите внимание на то, что HTML-дескрипторы самого элемента **в возвращаемый результат не включаются.**

В то же время

```
var text = $('li.dcell').text();  
console.log(text);
```

выведет текстовое содержимое всех выбранных элементов

## Изменение содержимого элементов

Для изменения содержимого элементов используются методы **html()** и **text()**

```
var htmlContent = $('li.dcell').html();
```

```
$('#myBasket').html(htmlContent); - изменит содержимое  
элемента с id=myBasket
```

```
$('li.dcell', '#row1').html(htmlContent); -
```

Этот сценарий изменяет HTML-содержимое тех принадлежащих классу dcell элементов li, которые являются потомками элемента с id=row1. Для получения содержимого, подлежащего вставке, используется метод html(), который считывает HTML-содержимое из первого элемента li.dcell.

## Изменение содержимого элементов с помощью функции

Передача методам `html ()` и `text ()` функции в качестве аргумента обеспечивает возможность динамического изменения содержимого элементов.

В обоих случаях аргументами функции являются **позиция** элемента в объекте jQuery и **текущее текстовое или HTML-содержимое**. Переменная `this` ссылается на объект `HTMLElement` соответствующий элементу, а **возвращаемое значение** функции содержит требуемый результат.

```
$('#label').text(function(index, currentValue) {  
return currentValue+"_"+index;  
});
```

**В этом сценарии текстовое содержимое элементов `label` изменяется в зависимости от значения аргумента `index`, указывающего позицию элемента в выбранном наборе**

## Работа с элементами формы

Для получения и изменения значений элементов формы (таких, как `input`) используется метод `val()`

**`val()`** Возвращает значение первого из элементов, содержащихся в объекте `jQuery`

**`val(значение)`** Изменяет значения всех элементов, содержащихся в объекте `jQuery`

**`val(функция)`** Изменяет значения всех элементов, содержащихся в объекте `jQuery`, с помощью функции

Пример использования метода `val ()` для получения значения первого из элементов, содержащихся в объекте `jQuery`.

Метод `each()` используется для перебора значений всех элементов `input`, содержащихся в документе.

```
$('#ul.drow input').each(function(index, elem){  
  console.log("Имя:" + " " + elem.name + " Значение: " + $(elem).val());  
});  
$('#ul.drow input').val('1');- устанавливает значение равное единице.
```

## Изменение значений элементов формы с помощью функции

Для изменения значений с помощью метода `val()` можно использовать также функцию. Аргументами этой функции являются позиция элемента в наборе и его текущее значение. Переменная **this** ссылается на объект `HTMLElement`, представляющий обрабатываемый элемент.

```
$('input').val(function(index, currentVal) {  
return currentVal+2;  
});
```

## Селекторы формы

- :button** Выбирает все кнопки (`input[type=submit]`, `input[type=reset]`, `input[type=button]` или `button`).
- :checkbox** Выбирает только элементы-флажки (`input[type=checkbox]`).
- :checked** Выбирает только отмеченные флажки или радиокнопки.
- :disabled** Выбирает только элементы форм, находящиеся в неактивном состоянии
- :enabled** Выбирает только элементы форм, находящиеся в активном состоянии.
- :file** Выбирает все элементы типа `file` (`input[type=file]`).
- :image** Выбирает изображения в формах (`input[type=image]`).
- :input** Выбирает только элементы форм (`input`, `select`, `textarea`, `button`).
- :password** Выбирает только элементы ввода пароля (`input[type=password]`).
- :radio** Выбирает только радиокнопки (`input[type=radio]`).
- :reset** Выбирает только кнопки сброса (`input[type=reset]` или `button[type=reset]`).
- :selected** Выбирает элементы `<option>`, которые были выделены.
- :submit** Выбирает кнопки отправки формы (`button[type=submit]` или `input[type=submit]`).
- :text** Выбирает только элементы ввода текста (`input[type=text]`).

Получить значения выбранного элемента списка  
`$( "select.foo option:selected").val();`

// другой способ получить значение  
`$( "select.foo" ).val();`

// Получить значение выбранного флажка  
`$( "input:checkbox:checked" ).val();`

// Получить значение выбранной радиокнопки

`$( "input:radio[name=bar]:checked" ).val();`  
`$( "[name='radioGroup']:checked" ).val();`

`var checkboxValues = $( "[name='checkboxGroup']:checked" ).map(  
function(){ return $(this).val(); }).toArray();`