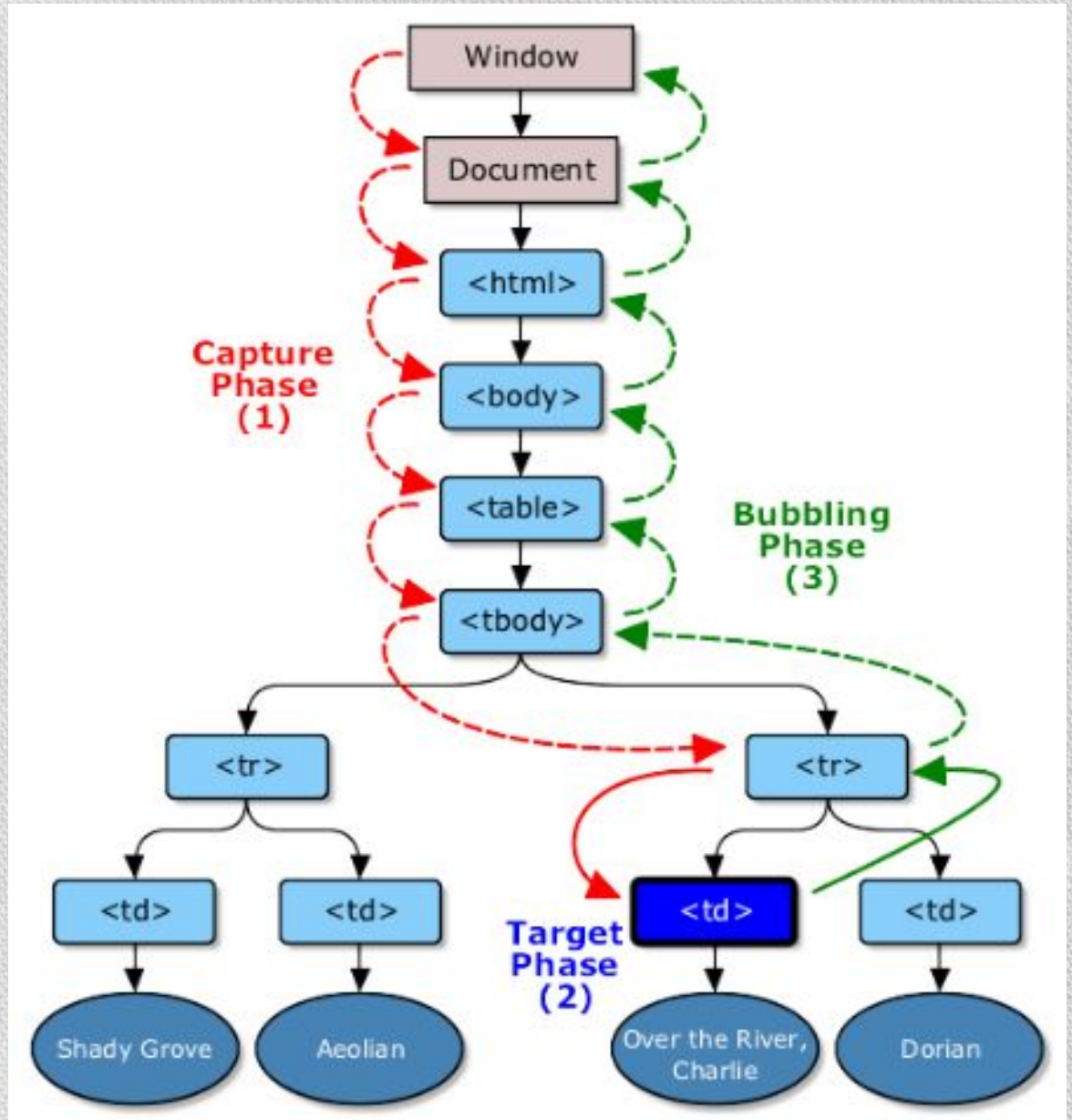
РАБОТА С DOM В JQUERY

Что такое DOM?



- HTML-документы имеют иерархическую структуру, представленную в DOM дереве. Узлы дерева представляют различные типы содержимого документа.
- В первую очередь, древовидное представление HTML-документа содержит узлы, представляющие элементы, такие как <a>, <p>, и узлы, представляющие строки текста.

Организация работы с DOM в jQuery

- jQuery—мощная библиотека JavaScript, ориентированная на упрощение взаимодействия с HTML через JavaScript.
- В jQuery предусмотрена работа с DOM-деревом при помощи специальных методов.

Функции фильтрации набора элементов

Функция	Описание
.eq(index)	Сокращает число совпавших элементов до одного. Аргументом является позиция элемента в наборе совпавших элементов.
.not(selector)	Удаляет элементы, соответствующие указанному выражению, из набора совпавших элементов.
.filter(function)	Удаляет все элементы, которые не удовлетворяют требованиям функции, из набора совпавших элементов.
.is(selector)	Проверяет текущий набор элементов на соответствие указанному выражению и возвращает true, если хотя бы один элемент соответствует выражению.
.slice(index, index)	Выделяет подмножество из набора совпавших элементов.

Методы фильтрации набора элементов: метод eq(index)

.eq(index)- сокращает число совпавших элементов до одного. Аргументом является позиция элемента в наборе совпавших элементов, начинается с 0 и продолжается до length-1. Указание некорректного индекса влечет за собой возвращение пустого набора элементов, а не нуля, поскольку запрос отфильтровывает все элементы, несоответствующие указанному индексу.

Методы фильтрации набора элементов: метод not(selector)

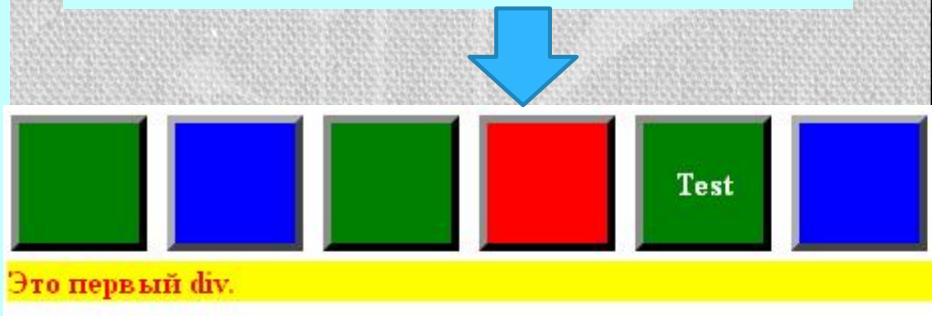
.not(selector)-удаляет элементы, соответствующие указанному выражению ,из набора совпавших элементов. Выражение может быть как отдельным селектором, так и сложной селекторной конструкцией.

Методы фильтрации набора элементов: метод is(n)

.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.");
            }
        });
    });
```

```
<body>
  <div></div>
  <div class="blue"></div>
  <div></div>
  <div class="red"></div>
  <div>
     <br /> <span>Test</span>
  </div>
  <div class="blue"></div>
  <p> </p>
</body>
```



Методы фильтрации набора элементов: методы filter()

.filter(selector)-удаляет все элементы, которые не соответствуют указанному выражению, из набора совпавших элементов.

```
$(“p”).filter(“.class1”).css(“background-color”, “Red”);
```

Simple

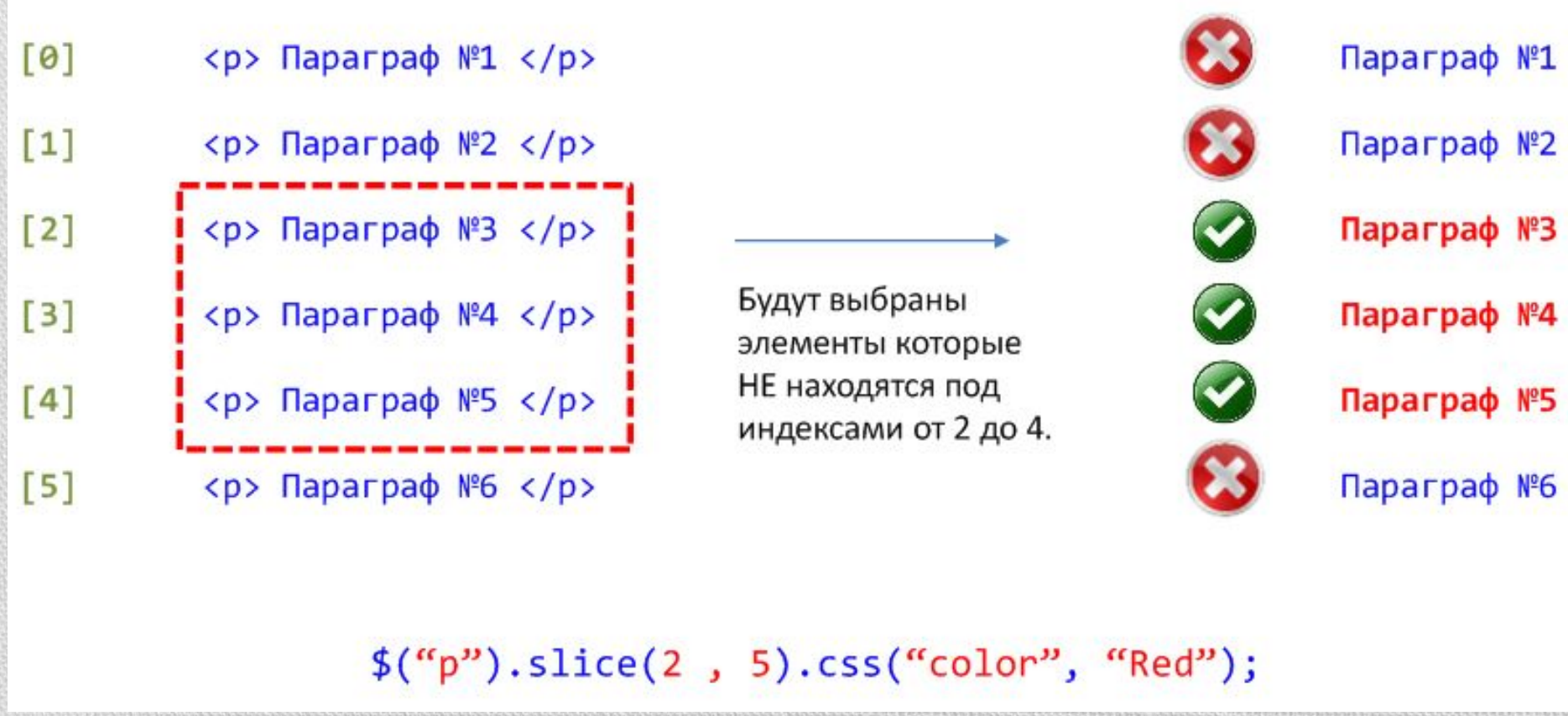
.filter(function). Если передать методу filter() некоторую функцию, он вызовет ее для каждого элемента в наборе и исключит те элементы, для которых эта функция вернет значение false.

```
$(“p”).filter(
        function(i) {
                return i != 3 && i< 7;
}
).css(“background-color”, “Red”);
```

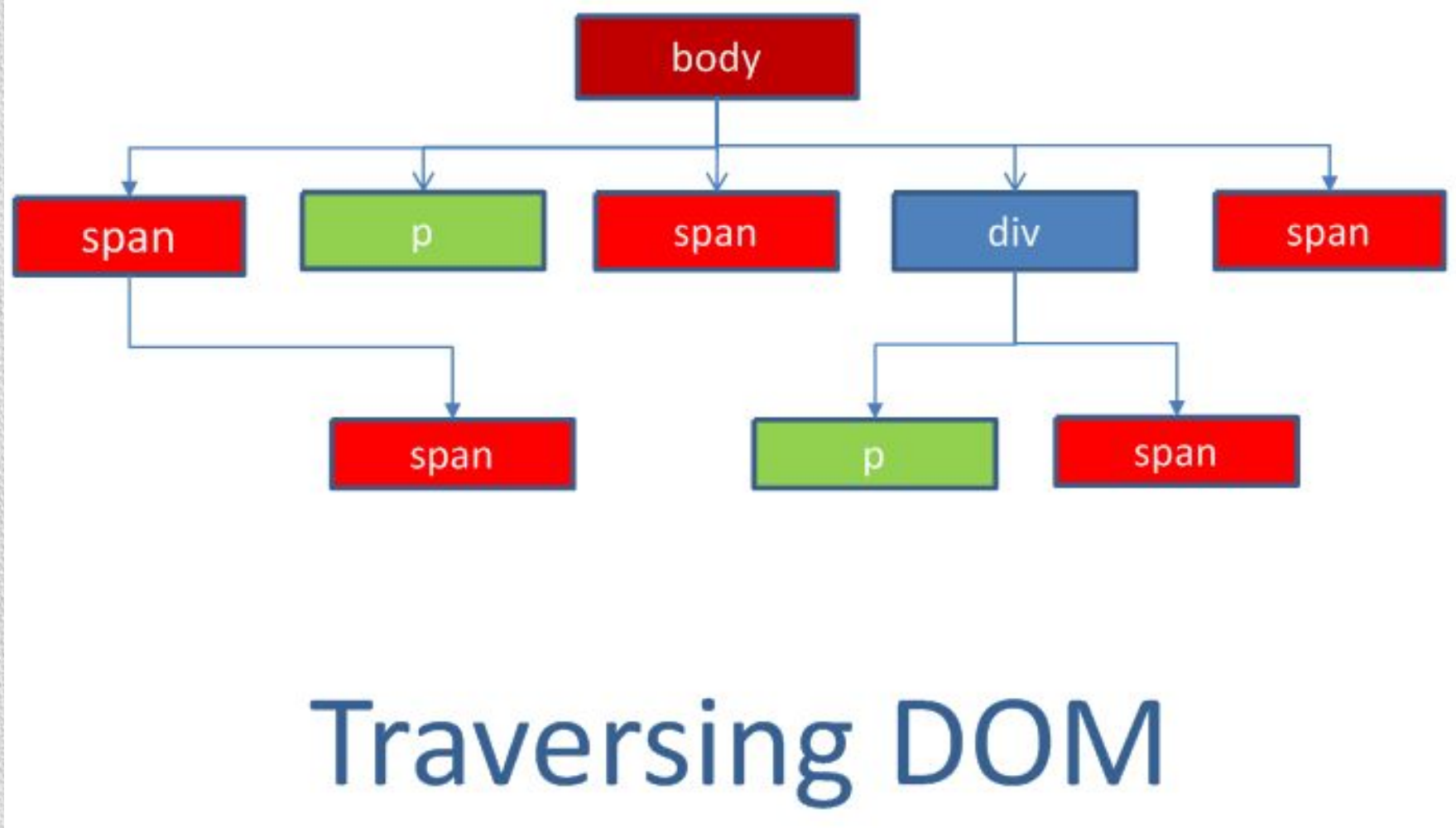
Simple

Методы фильтрации набора элементов: метод slice(start, stop)

.slice(start, stop) - выделяет подмножество из набора совпавших элементов. Ведет себя точно также как и встроенный метод разделения массивов (допускаются отрицательные значения).Не включая последний индекс, но включая первый.



Перемещение по DOM-дереву

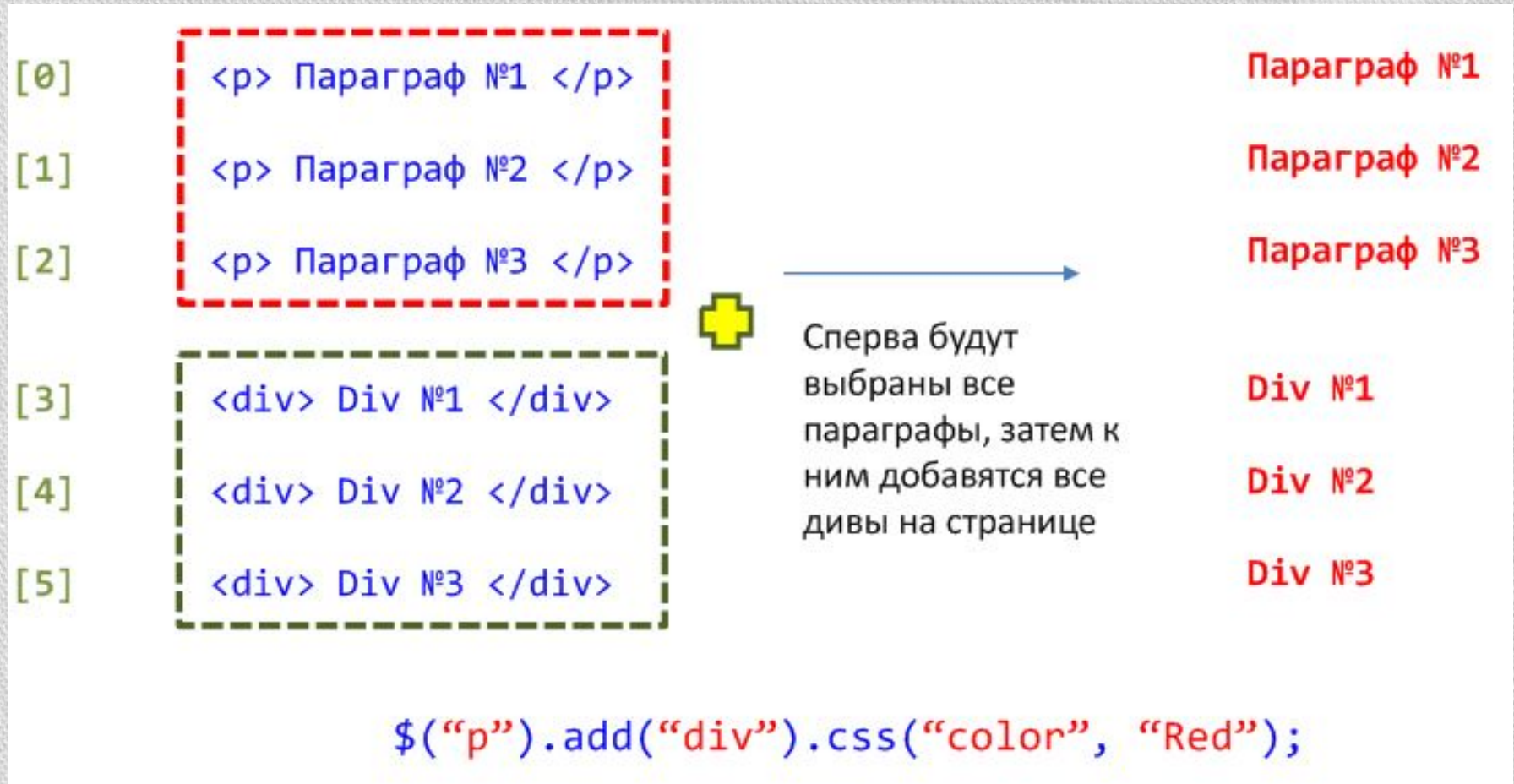


Функции поиска по DOM в jQuery

Функция	Описание
.add(выражение)	добавляет в уже существующий набор дополнительные элементы, которые удовлетворяют указанному выражению.
.find(выражение)	отыскивает дочерние элементы, которые удовлетворяют указанному выражению.
.contents()	поиск всех дочерних узлов в наборе совпавших элементов (включая текстовые) или в содержимом документа, если он является фреймом.
.children()	получает набор элементов, содержащий всех непосредственных уникальных потомков для каждого совпавшего элемента.
.sibling()	получает набор элементов, содержащий все уникальные родственные элементы для набора совпавших элементов.
.andSelf()	Добавление предыдущего набора к текущему набору.
.end()	Отменяет последнее деструктивное действие.

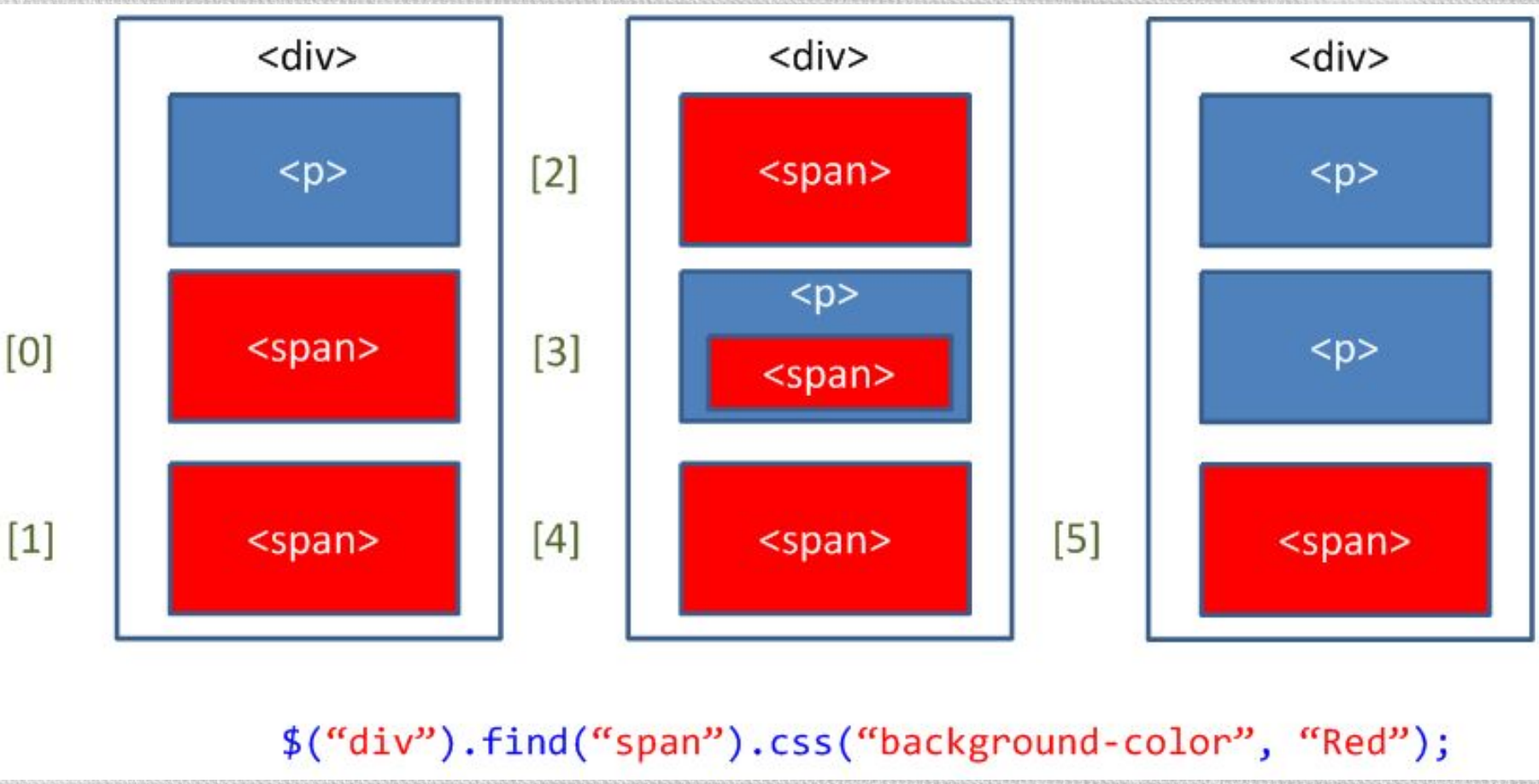
Перемещение по DOM-дереву: метод add(selector)

.add (выражение)-добавляет в уже существующий набор дополнительные элементы, которые удовлетворяют указанному выражению.



Перемещение по DOM-дереву: метод find(selector)

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

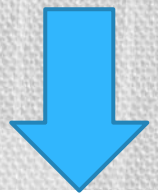


Перемещение по DOM-дереву: метод contents()

.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");

    });
```

```
<body>
    <div id="content">
        lorem ipsum
        <span>lorem ipsum in span</span>
    </div>
</body>
```

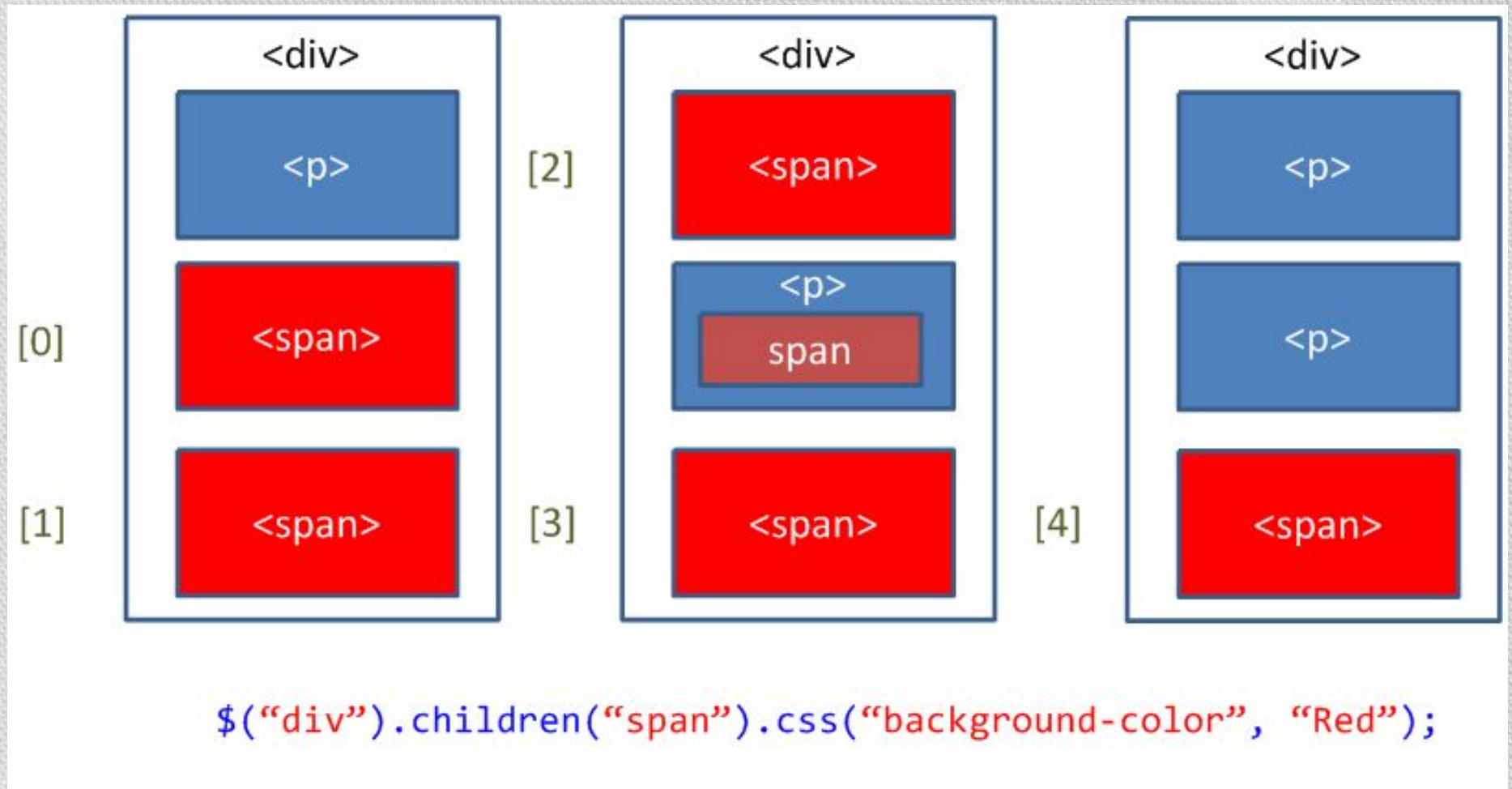


lorem ipsum

lorem ipsum in span

Перемещение по DOM-дереву: метод children(selector)

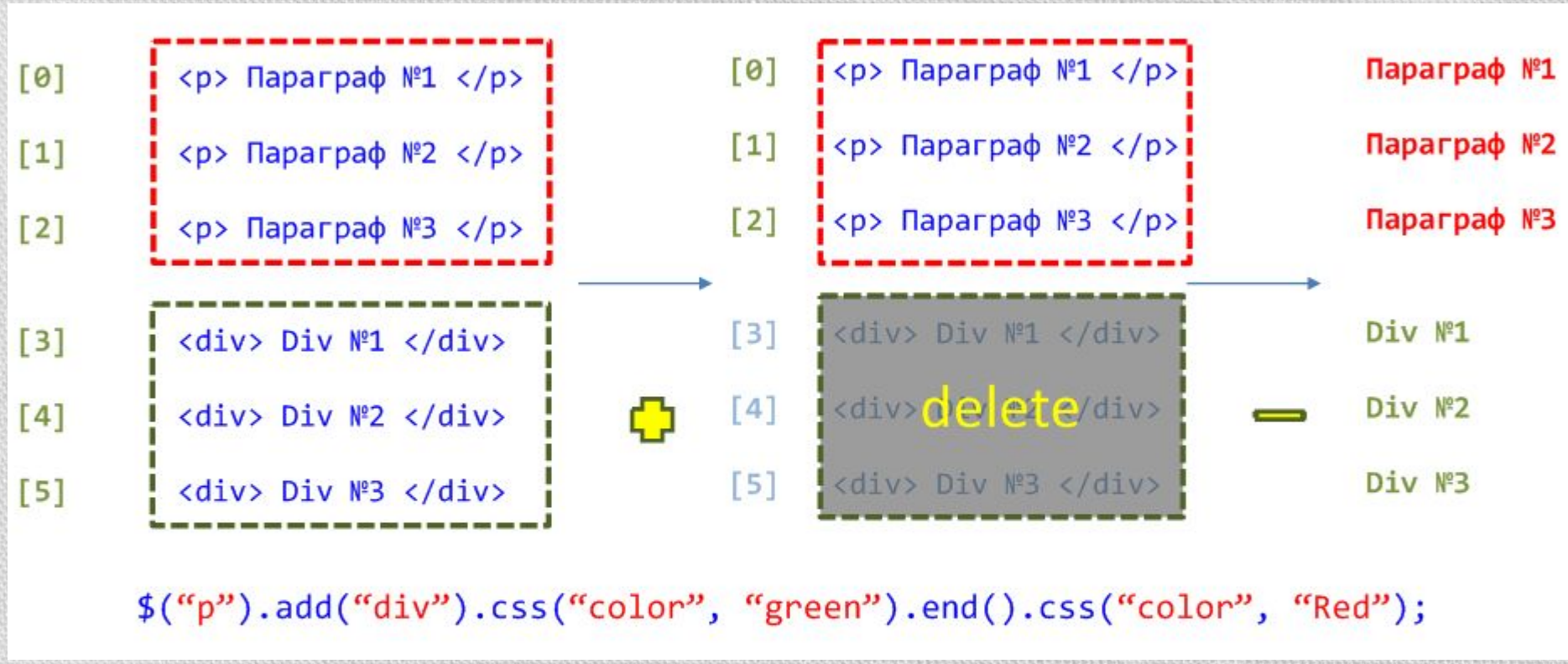
.children (выражение)-получает набор элементов, содержащий всех непосредственных уникальных потомков для каждого совпавшего элемента.




```
$("div").children("span").css("background-color", "Red");
```

Перемещение по DOM-дереву: метод end()

.end ()–отменяет последний фильтр, возвращая тем самым набор элементов к его предыдущему состоянию (до фильтрации)



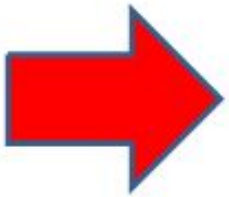
Функции поиска по DOM в jQuery

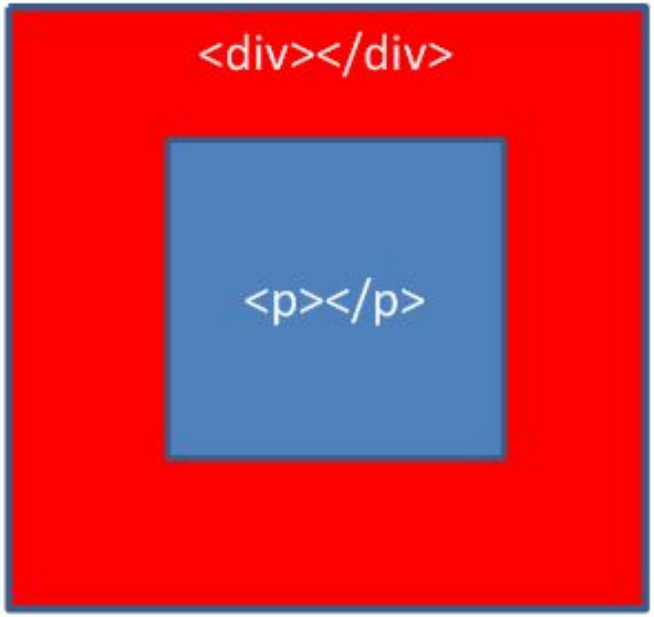
Функция	Описание
.parent(выражение)	Получает прямого “родителя” элемента.
.parents(выражение)	Получает набор элементов, содержащий уникальных “родителей” для совпавших элементов (кроме корневого элемента).
.closest()	Получает набор, содержащий ближайшие родительские элементы, которые удовлетворяют указанному селектору, включая начальный элемент.
.offsetParent()	Возвращает коллекцию jQuery с “родителем” по позиционированию” первого совпавшего элемента.
.next(), .prev()	Получает набор элементов, содержащий уникальные последующие(предыдущие) родственные элементы для всех элементов существующего набора.
.nextAll(), .prevAll()	Поиск всех родственных элементов после текущего элемента (перед текущим элементом).
.nextUntil(селектор), .prevUntil(селектор)	Поиск всех родственных элементов после текущего элемента (перед текущим элементом) доуказанного селектора.

Методы фильтрации набора элементов: метод parent()

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

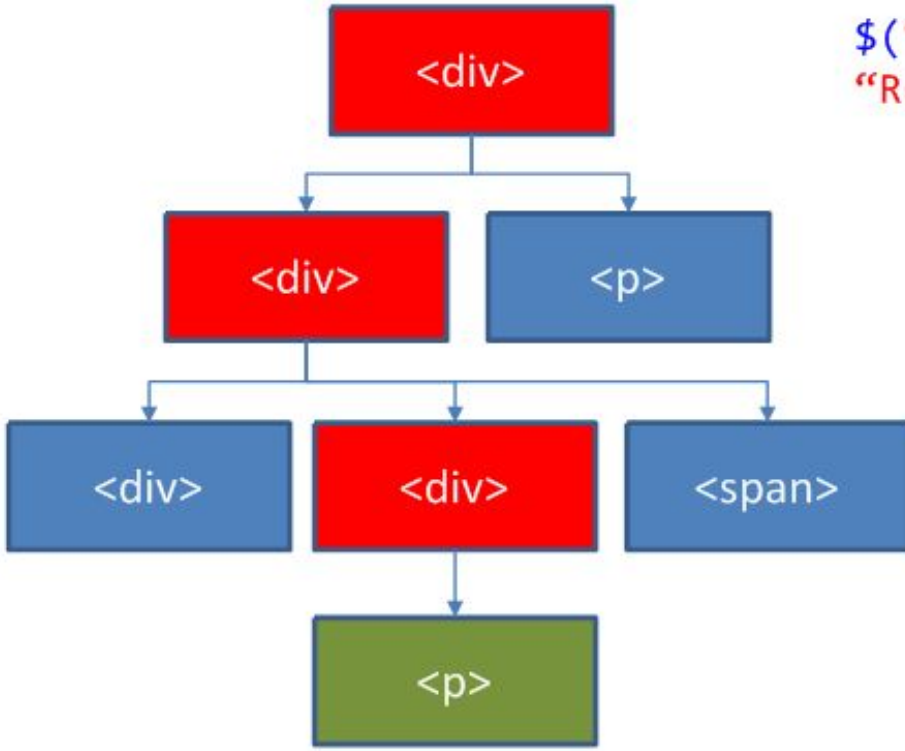
```
$("p").parent()
.css("background-color", "Red");
```





Методы фильтрации набора элементов: метод parents()

.parents () -поиск всех предков выбранных элементов. Будут выбраны не только прямые родители, но и прародители, прапрародителии так далее до начала DOM-дерева

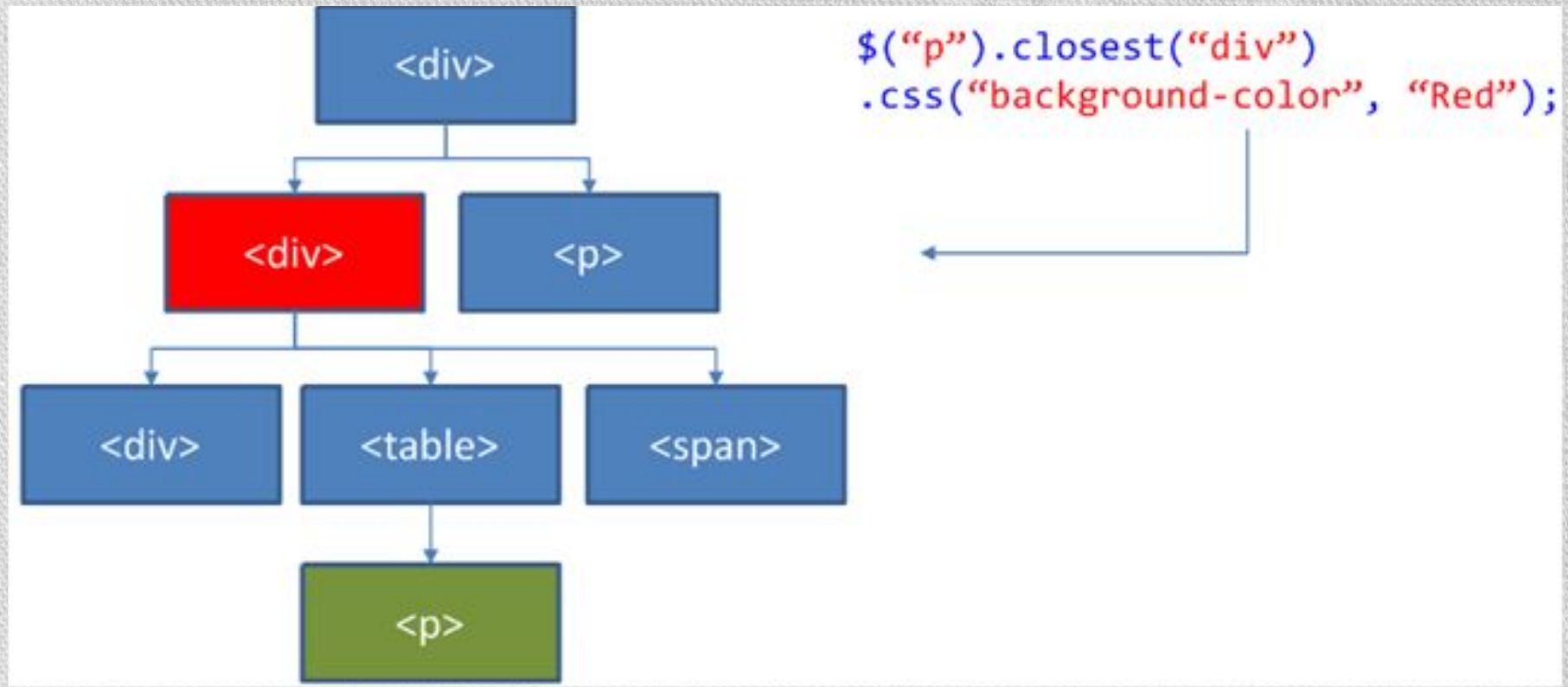



```
$("p").parents().css("background-color",
"Red");
```

Simlpe

Методы фильтрации набора элементов: метод closest(selector)

.closest (выражение) -начинает поиск непосредственно с выбранного (выбранных) элемента и движется вверх по иерархии. Двигаясь вверх по иерархии останавливает поиск после первого подходящего элемента (после чего начинает осуществлять поиск для следующего выбранного элемента). Поэтому, находит не более одного элемента для каждого из выбранных.

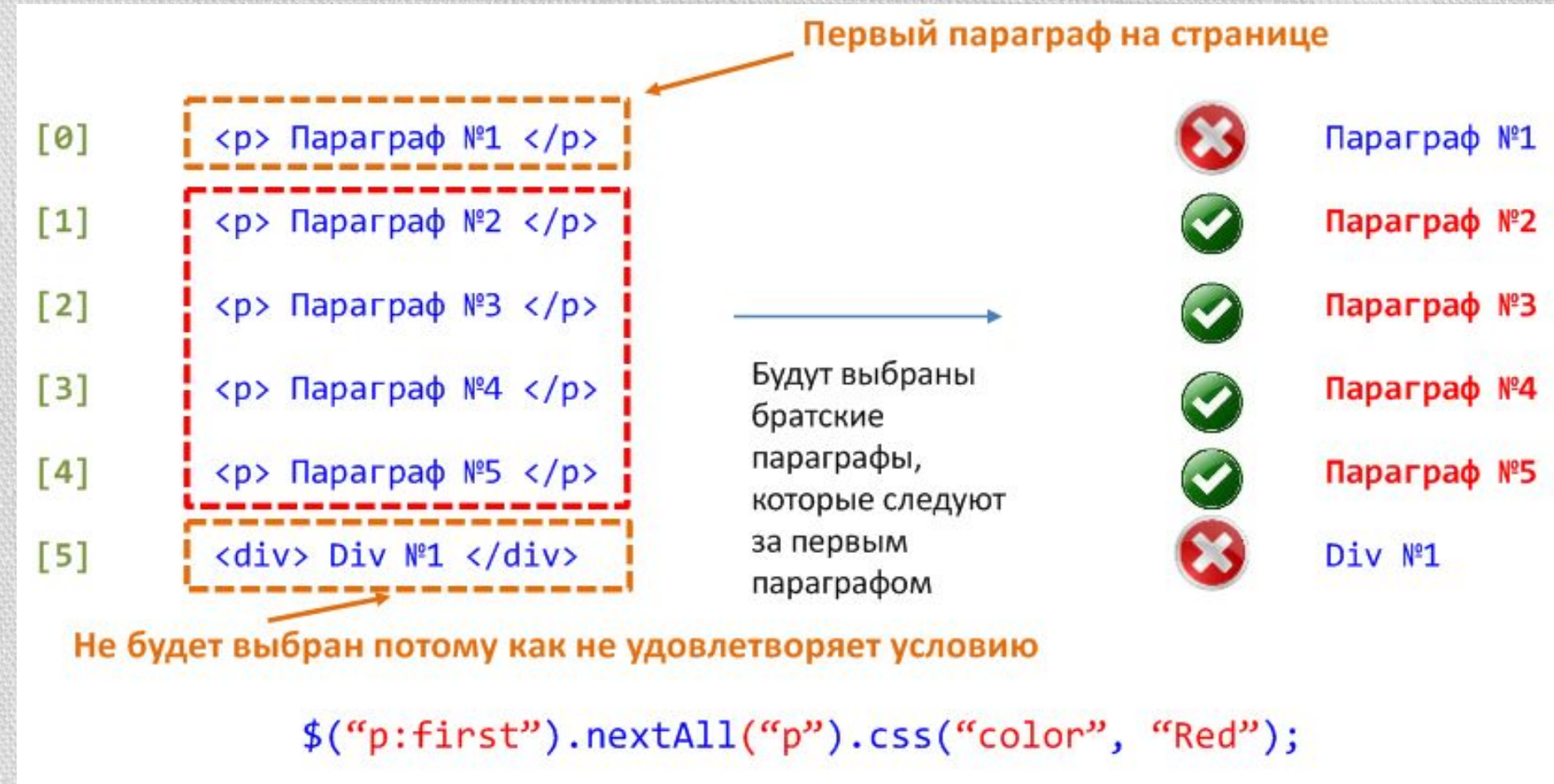


Методы фильтрации набора элементов: метод next()

.next () -получает набор элементов, содержащий только соседние элементы, следующие непосредственно за указанным элементом.

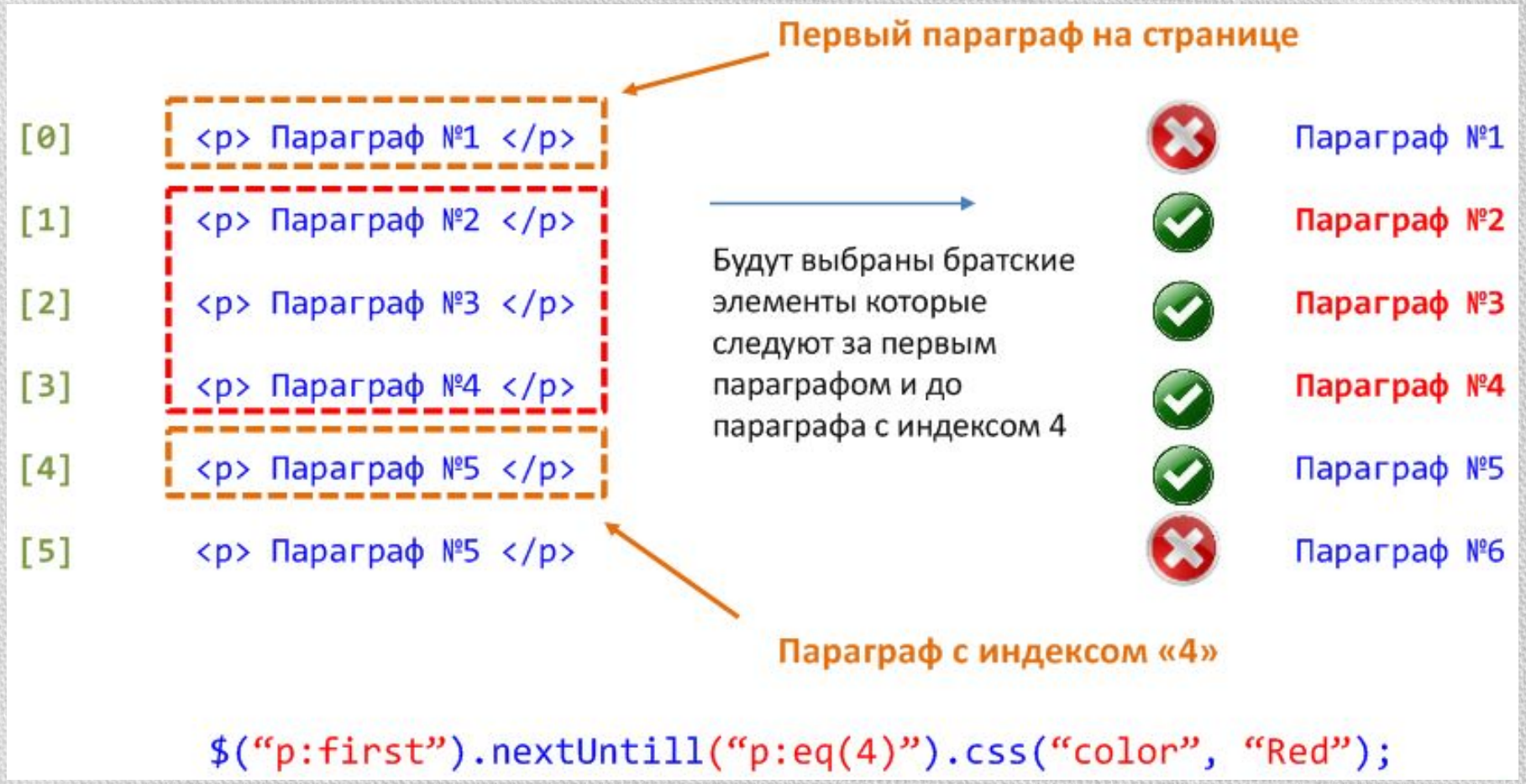
Методы фильтрации набора элементов: метод nextAll()

.nextAll () –отыскивает все последующие братские элементы, которые соответствуют селектору и идут после выбранного элемента.



Методы фильтрации набора элементов: метод nextUntil()

.nextUntil() –получает всех последующих братьев каждого элемента до, но не включая элемент, соответствующий переданному селектору, узлу DOM или объекту jQuery.

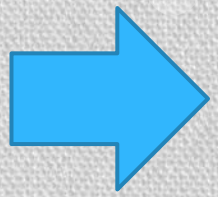


Методы фильтрации набора элементов: метод prev()

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

```
$(function () {
        $("p").prev().css("background",
"yellow");
});
```

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



Hello Again (span)

Hello (paragraph)

Hello Again (span)

And Again

Hello Again (span)

Hello (paragraph)

Методы фильтрации набора элементов: метод prevAll()

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");
```

```
<body>
    <p>Параграф №1</p>
    <p>Параграф №2</p>
    <p class="class1">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>
    <div>А это простой Div</div>
</body>
```

```
<style>
        body {
            background-color: Silver;
            color: White;
        }

        p {
            padding: 10px;
            font-weight: bold;
            background-color: Gray;
        }
    </style>
```

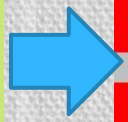
Методы фильтрации набора элементов: метод prevUntil(selector)

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

```
 $(function () {
$(".class2").prevUntil(".class1").css("backgr
ound-color", "Red");
        });
```

```
<body>
    <p>Параграф №1</p>
    <p>Параграф №2</p>
    <p class="class1">Параграф №3</p>
    <p>Параграф №4</p>
    <p>Параграф №5</p>
    <p>Параграф №6</p>
    <p class="class2">Параграф №7</p>
    <p>Параграф №8</p>
    <p>Параграф №9</p>
    <p>Параграф №10</p>
    <div>А это простой Div</div>
</body>0
```

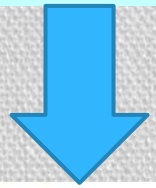
```
<style>
        body {
            background-color: Silver;
            color: White;
        }

        p {
            padding: 10px;
            font-weight: bold;
            background-color: Gray;
        }
    </style>
```



Методы фильтрации набора элементов: метод siblings()

Метод siblings() - получение всех родственных элементов.

```
$(function () {

        var len = $(".highlight")
            .siblings()
            .css("color", "red")
            .length;

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

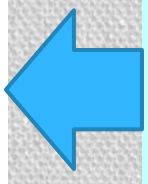


- One
- Two
- Three
- Four

- Five
- Six
- Seven

- Eight
- Nine

Unique siblings: 4

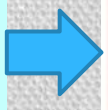


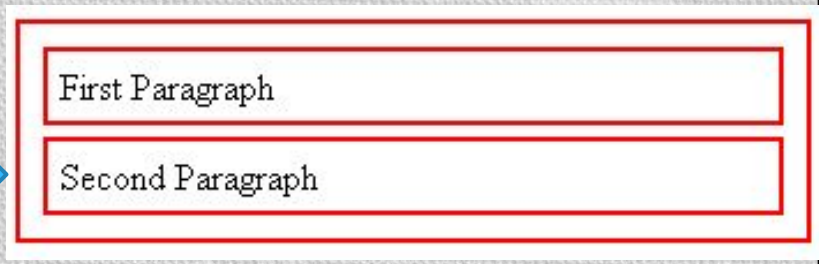
```
<body>
    <ul>
        <li>One</li>
        <li>Two</li>
        <li class="highlight">Three</li>
        <li>Four</li>
    </ul>

    <ul>
        <li>Five</li>
        <li>Six</li>
        <li>Seven</li>
    </ul>

    <ul>
        <li>Eight</li>
        <li class="highlight">Nine</li>
    </ul>
    <p>Unique siblings: <b></b></p>
</body>
```

Методы фильтрации набора элементов: метод AndSelf()

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
```






```
<body>
  <div>
    <p>First Paragraph</p>

    <p>Second Paragraph</p>
  </div>
</body>
```

```
<style>
      p, div {
          margin: 5px;
          padding: 5px;
      }

      .border {
          border: 2px solid red;
      }
  </style>
```

Методы фильтрации набора элементов: метод find()

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"
            });
    });
```

```
<style>
    p {
        font-size: 20px;
        width: 200px;
        cursor: default;
        color: blue;
        font-weight: bold;
        margin: 0 10px;
    }

    .highlight {
        background: yellow;
    }
</style>
```

When *the* day is *short*
find *that* which *matters*
to you or *stop* believing

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

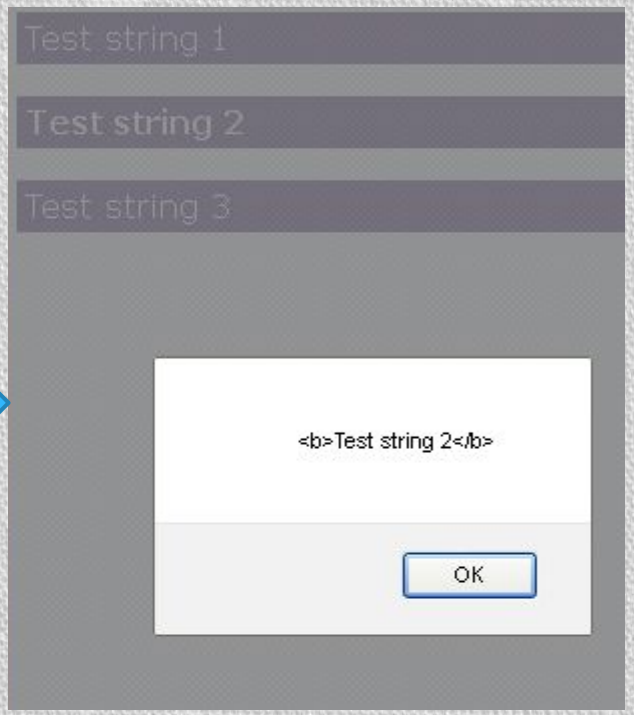
Методы вставки элементов в DOM: метод Html()

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());
        });
    });
```

```
<body>
    <p>
        Test string 1
    </p>
    <p>
        <b>Test string 2</b>
    </p>
    <p>
        Test string 3
    </p>
</body>
```

```
<style>
        p {
            background-color: #bdb1c9;
            color: White;
            padding: 4px;
            font-family: Verdana;
            cursor: pointer;
        }
    </style>
```



Пример запись в документ

Методы вставки элементов в DOM: метод Text()

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());
        });

    });
```

```
<body>
    <p>
        <b>Test</b> string 1
    </p>
    <p>
        Test string 2
    </p>
</body>
```

Пример запись в документ

Методы вставки элементов в DOM: метод Append() , AppendTo()

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

```
$(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".
      });
```


```
<body>
   <p id="first">
      hello
   </p>
   <table border="1">
      <tr>
         <td>1</td>
         <td>2</td>
      </tr>
   </table>
   <br />
   <div class="test">1 </div>
   <div class="test">2 </div>
   <div class="test">3 </div>
</body>
```

Методы вставки элементов в DOM: метод Prepend() , PrependTo()

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".
    });
```


```
<body>
    <p id="first">
        hello
    </p>
    <table border="1">
        <tr>
            <td>1</td>
            <td>2</td>
        </tr>
    </table>
    <br />
    <div class="test">1 </div>
    <div class="test">2 </div>
    <div class="test">3 </div>
</body>
```

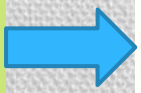
Методы вставки элементов в DOM: метод wrap(), empty(), remove()

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(); });
    });
```

```
<body>
    <span>Paragraph</span>
    <p>Hello world</p>
    <input type="button" value="Clear"
id="clearBtn" />
    <input type="button" value="Remove"
id="removeBtn" />
</body>
```

```
<style>
    div {
        background-color: Blue;
         padding: 10px;   }
 span {background-color: White
</style>
```



Задания

Задание 1

Откройте файл “C_Task1.html”. В этом файле Вам нужно выбрать средствами библиотеки JQuery неактивные кнопки (button[disabled]) и вписать в теги span, которые идут сразу же после этих кнопок текст “This button is disabled”.

Result

Задание 2

Откройте файл “H_Task1.html”. В этом файле Вам нужно вставить в див неупорядоченный список на 3 элемента средствами библиотеки JQuery.

Result

Задание 3

Откройте файл “H_Task2.html”. В этом файле Вам нужно выбрать параграф и затем закрасить его родительский элемент красным цветом. библиотеки JQuery.

Result