



# Полный курс

Lesson 1

Введение в jQuery

# Введение в jQuery

## Что такое jQuery?

jQuery - быстрая, компактная, мощная JavaScript-библиотека.

С ее помощью можно взаимодействовать с HTML-документом, обрабатывать события, реализовывать анимационные эффекты и работать с AJAX намного проще благодаря удобному API, который к тому же еще и кроссбраузерный.

## Кто использует jQuery?



# Введение в jQuery

## История создания jQuery

Библиотека была разработана молодым разработчиком Джоном Резигом.

В 2006 году jQuery была представлена общественности на компьютерной конференции «BarCamp» в Нью-Йорке.



На данный момент развитием и поддержкой фреймворка занимается команда jQuery Team под руководством Джона Резига.

# Подключение jQuery

## Как подключить библиотеку jQuery?

Подключить библиотеку jQuery можно двумя способами:

1. Разместить ссылку на библиотеку через jQuery CDN:

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

2. Разместить ссылку на локальную копию библиотеки:

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

Официальный сайт библиотеки: <http://jquery.com/>

# Работа с jQuery

## Функция jQuery()

Функция `jQuery()` – основная функция в библиотеке, которая позволяет производить поиск элементов на веб-странице, создавать новые элементы, манипулировать ими и многое другое. Возвращает объект jQuery.

Для упрощения написания кода используют сокращенную запись функции `jQuery()` – функцию `$()`.

В качестве аргументов она принимает селекторы (элементы), над которыми следует выполнить какие-то действия.

`$('p');` // выбор всех параграфов на веб-странице.

# Работа с jQuery

## Подготовка работы с веб-страницей

Прежде чем начинать выполнять JavaScript-код, нужно быть уверенным в загрузке всех элементов страницы (загрузке DOM-дерева). В классическом JavaScript-коде чаще всего отслеживают полную загрузку страницы событием `onload`:

```
window.onload() = function(){  
    // тело функции  
};
```

В jQuery используют событие `ready`, которое происходит в момент готовности DOM-дерева:

```
$(document).ready(function () {  
    // тело функции  
});
```



# Базовые селекторы

## Понятие «селектор»

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

Селекторы позволяют находить элементы по различным признакам: значению атрибутов, содержимому элементов, родительским элементам, дочерним элементам, порядковым номерам, ну и конечно по именам классов, идентификаторов и/или тегов.

Например выражение `$("div")` осуществляет поиск всех `div`-элементов на странице.

# Базовые селекторы

## Выборка по имени тега

<script>

...

**\$("a")**... прочие действия

...

</script>

$\$(\text{'a'})$  - соответствует всем элементам с именем тега "a".

<a href="#">...</a>



# Базовые селекторы

## Выборка по имени класса

```
<script>
```

```
...
```

```
$(".yellow")... прочие действия
```

```
...
```

```
</script>
```

`$(".yellow")` – выбор элементов, с именем класса «yellow».

```
<div class="yellow">...</div>
```

```
<div class="green">...</div>
```

```
<div class="yellow">...</div>
```



# Базовые селекторы

## Выборка по идентификатору

```
<script>
```

```
...
```

```
$("#reg_form")... прочие действия
```

```
...
```

```
</script>
```

`$("#reg_form")` – выбор элемента, с именем идентификатора «reg\_form».



```
<div id="reg_form">...</div>
```

# Базовые селекторы

## Таблица выборов элементов

В качестве аргумента селекторной функции можно передавать любые селекторы из CSS.

Конструкция	Описание	Пример
*	Выбор всех элементов на странице.	\$("*")
\$("E F")	Соответствует всем элементам с именем тега F, вложенным в элемент с именем тега E.	\$("div a")
\$("E,G,H")	Соответствует всем элементам с именами тегов E, G, H.	\$("div, span, .post")
\$("E > F")	Соответствует всем элементам с именем тега F, являющимися прямыми потомками элементов с именем тега E.	\$("div > a")

# Базовые селекторы

## Таблица выборов элементов

Конструкция	Описание	Пример
<code>\$( "E + F" )</code>	Соответствует всем элементам с именем тега F, которым непосредственно предшествует элемент E на том же уровне вложенности.	<code>\$( "p + a " )</code>
<code>\$( "E ~ F" )</code>	Соответствует всем элементам с именем тега F, которым предшествует элемент E на том же уровне вложенности.	<code>\$( "span ~ p" )</code>

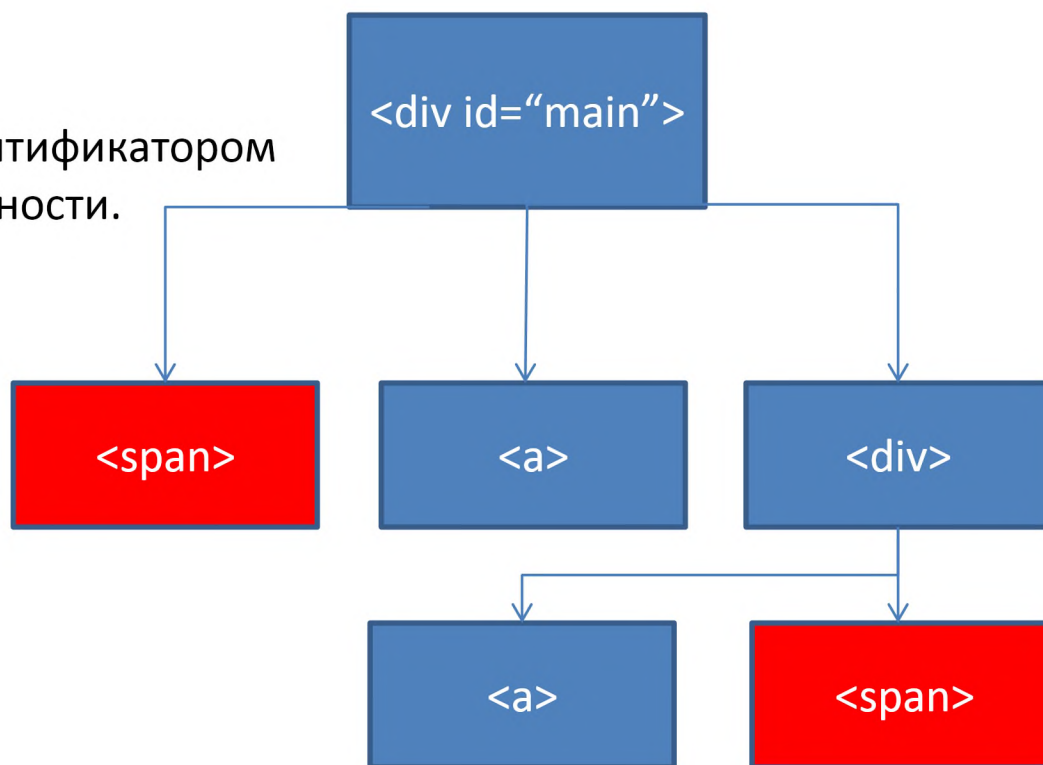


# Примеры базовых селекторов

## Селектор \$("E F");

```
$("#main span").css("background-color", "red");
```

Будут выбраны все теги "span",  
которые находятся в теге с идентификатором  
"main" на любом уровне вложенности.

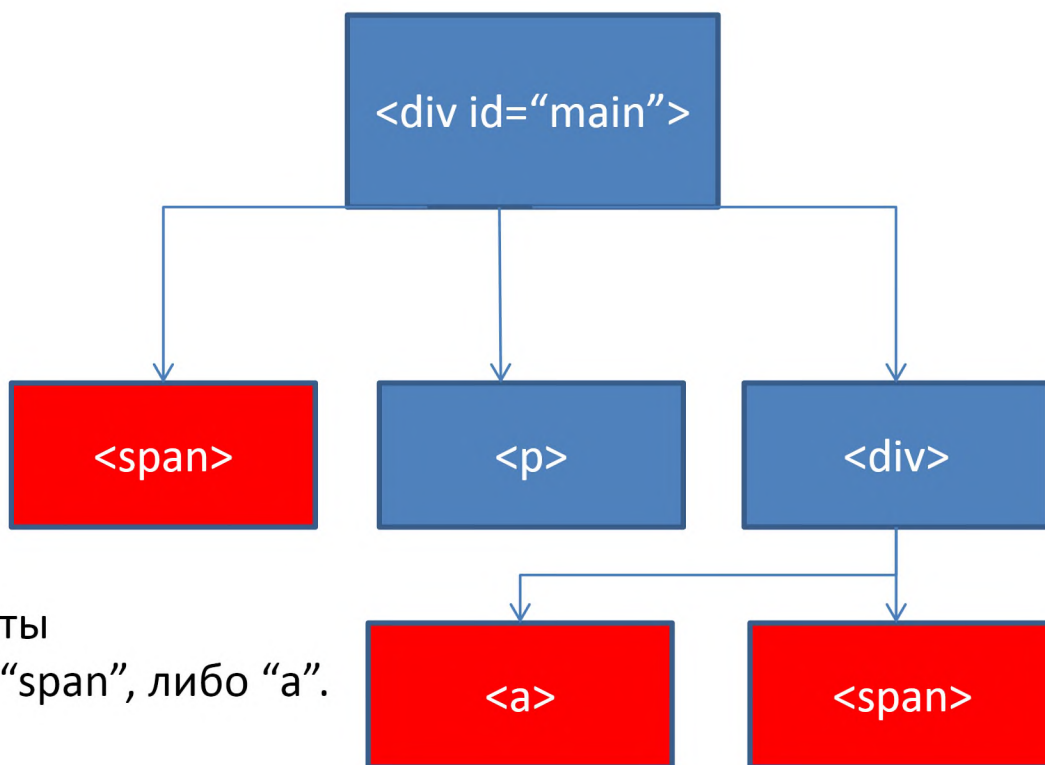




# Примеры базовых селекторов

Селектор `$("E, G");`

`$("span, a").css("background-color", "red");`

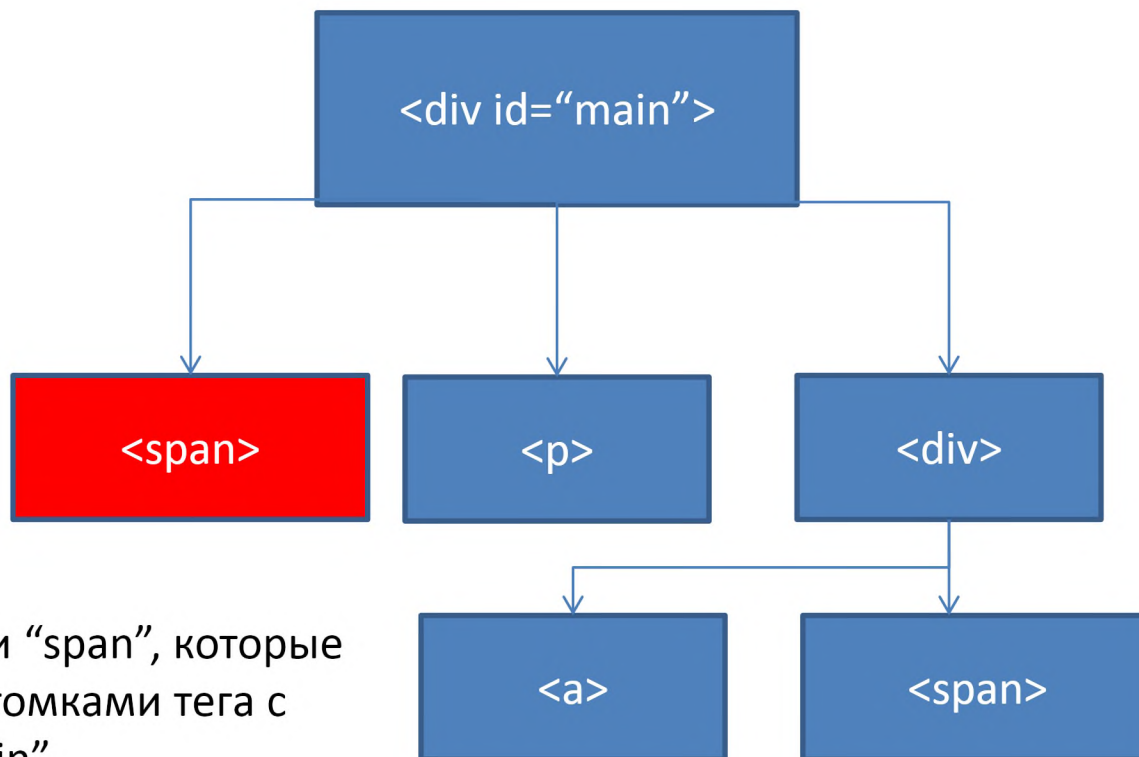


Будут выбраны все теги элементы  
которые являются либо тегами "span", либо "a".

# Примеры базовых селекторов

Селектор `$("E > F");`

`$("#main > span").css("background-color", "red");`

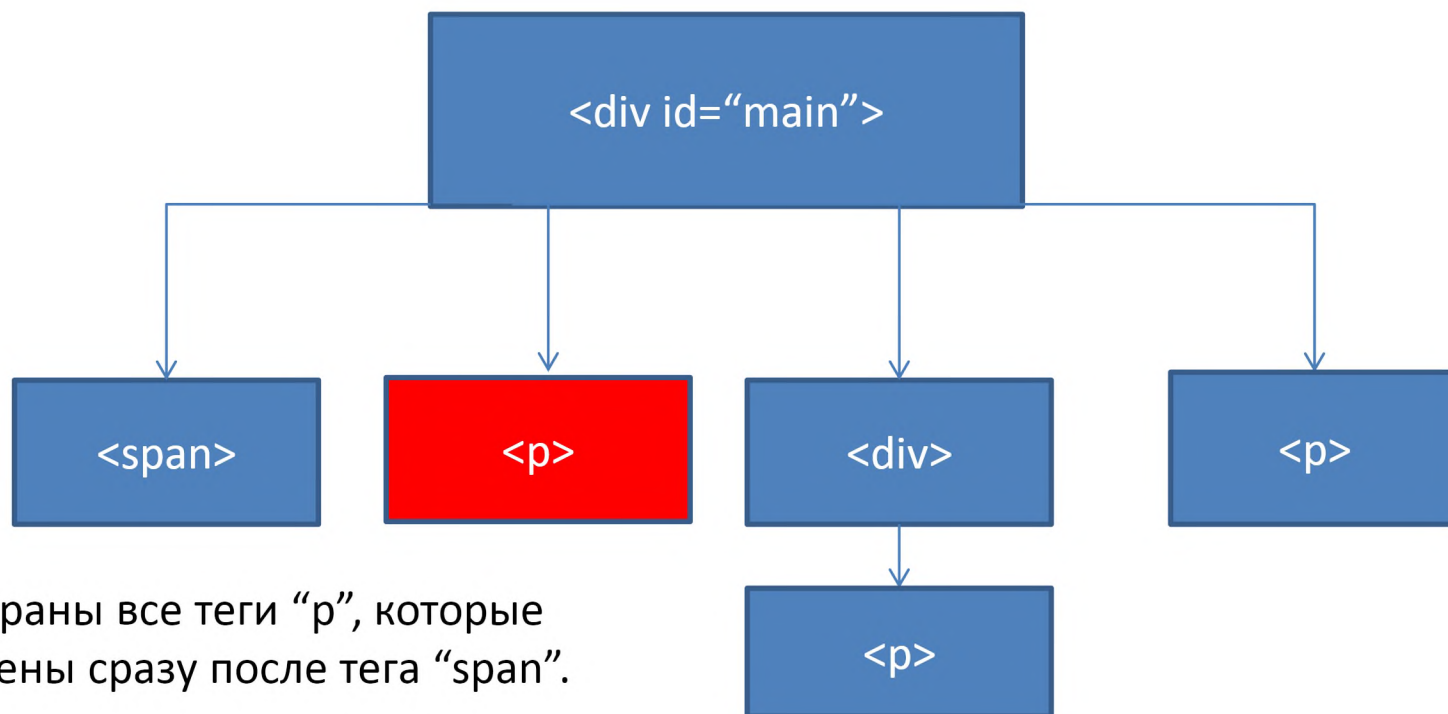


Будут выбраны все теги “span”, которые являются прямыми потомками тега с идентификатором “main”.

# Примеры базовых селекторов

Селектор `$("E + F");`

`$(" span + p ").css("background-color ", "red");`

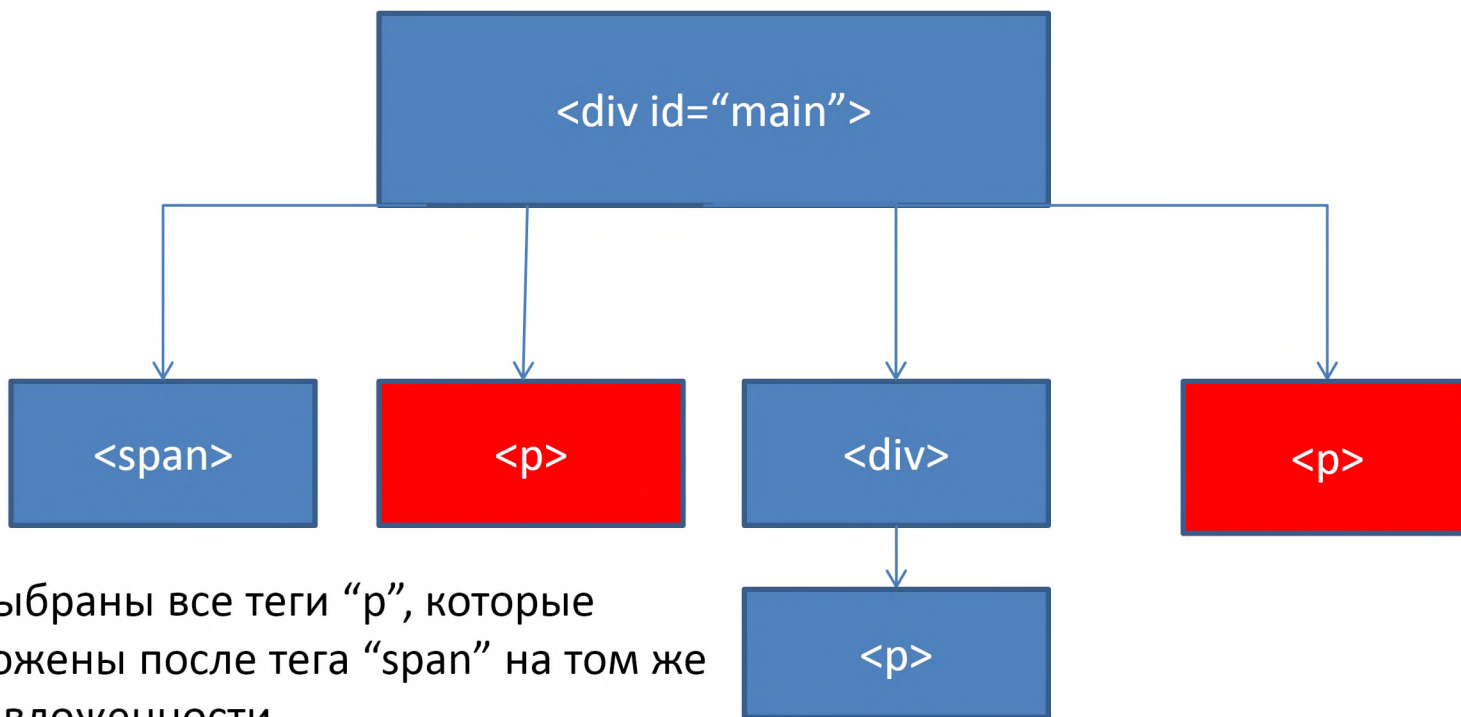


Будут выбраны все теги “p”, которые расположены сразу после тега “span”.

# Примеры базовых селекторов

Селектор `$("E ~ F");`

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



Будут выбраны все теги “p”, которые расположены после тега “span” на том же уровне вложенности.