



WEB разработка

jQuery въведение

Съдържание

- jQuery?
- Как да добавим jQuery към проекта си?
- Ресурси
- Селектиране и промяна на DOM
- Обхождане на DOM
- Добавяне/премахване на елементи
- jQuery обекти и събития

jQuery?

jQuery?

- Не е единствената JS библиотека, но е
- Най-използваната JS open-source библиотека
- Осигурява cross-browser съвместимост
- Улеснен синтаксис за
 - по-лесно обхождане и селектиране DOM елементите
 - създаване на анимации
 - управляване събития

jQuery?

- На какво дължи популярността си?
 - Лесна за изучаване
 - Лесна за разширяване
 - Лека
 - Много хора допринасят за развиването и поддръжката ѝ

!Не е достатъчна - няма да минете без vanilla JS!



Добавяне на jQuery

Добавяне на jQuery

1. Добавяме файловете в проекта си.
2. Използваме cdn

Ресурси

<https://jquery.com/>

Zak Ruvalcaba, Anne Boehm - **Murach's jQuery, 2nd edition** (Training & Reference) - 2015

Селектори и промяна на DOM елементи

Селектори и промяна на DOM елементи

Селекторите са подобни на JS селекторите

`$(selector) = jQuery(selector)`

//by tag

`$("div")`

//by id

`$("#navigation")`

//by class

`$(".menu-item")`

//by combination of selectors

`$("ul.menu li")`

Селектори и промяна на DOM елементи

The syntax for calling a jQuery method

`$\$("selector").methodName(parameters)$`

Some common jQuery methods

Method	Description
<code>val()</code>	Get the value of a text box or other form control.
<code>val(value)</code>	Set the value of a text box or other form control.
<code>text()</code>	Get the text of an element.
<code>text(value)</code>	Set the text of an element.
<code>next([type])</code>	Get the next sibling of an element or the next sibling of a specified type if the parameter is coded.
<code>submit()</code>	Submit the selected form.
<code>focus()</code>	Move the focus to the selected form control or link.

Обхождане на DOM дървото

Обхождане на DOM дървото next/prev

- Използваме свойствата на DOM елементите

Next and previous siblings

Parents and children

`jQuery.next()`

`jQuery.prev()`

Връща следващия

предишния

sibling

!!!☺)) Връща HTML – елемент – не [text] node☺))

Обхождане на DOM дървото parent/parent(selector)

`jQuery.parent()`

Връща родителския елемент на даден елемент

`jQuery.parents(selector)`

Връща първия родител, който съответства на селектора

Обхождане на DOM дървото parent/parents(selector)

```
<div id="wrapper">
  <ul id="items-list">
    <li>Item 1</li>
    <li>Item 2</li>
    <li class="special">Item 3</li>
    <li>Item 4</li>
  </ul>
</div>

<script type="text/javascript">
  var $node = $(".special");
  console.log($node.parent().attr("id"));
  console.log($node.parents("div").attr("id"));
  console.log($node.parents("#wrapper").attr("id"));
</script>
```

items-list

wrapper

wrapper



Добавяне/премахване на DOM елементи

Добавяне на DOM елементи

jQuery.appendTo()/prependTo()

```
$("<ul><li>Hello</li></ul>").appendTo("body");
```

jQuery.append()/prepend()

```
$("body").prepend("<h1>header</h1>");
```

Добавяне на DOM елементи

Създаваме лесно нови елементи

```
var $divElement = $('

');


```

```
var $anotherDivElement = $('
```

Премахване на DOM елементи

```
<body>
  <div>
    <p>Red</p>
    <p>Green</p>
  </div>

  <script type="text/javascript">

    //$( 'p' ).remove();

  </script>
```

```
▼ <body>
  ▼ <div> == $0
    <p>Red</p>
    <p>Green</p>
  </div>
  <script type="text/javascript">

    //$( 'p' ).remove();

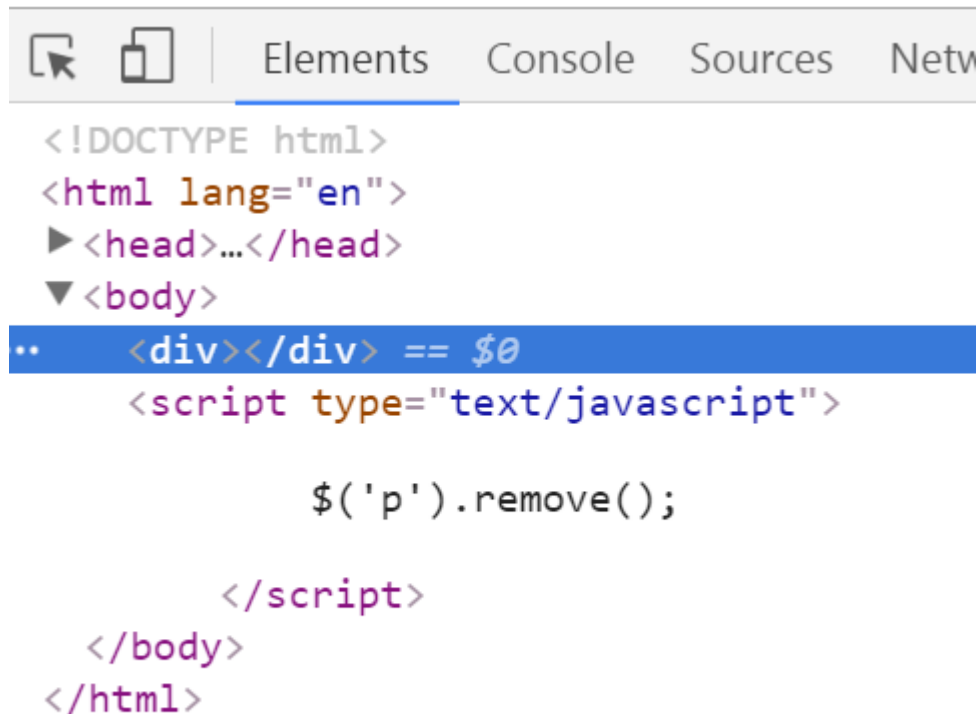
  </script>
</body>
</html>
```

html body div

Премахване на DOM елементи

```
</head>
<body>
  <div>
    <p>Red</p>
    <p>Green</p>
  </div>

  <script type="text/javascript">
    $('p').remove();
  </script>
</body>
</html>
```



The screenshot shows a web browser's developer tools interface. The 'Elements' tab is active, displaying the DOM tree. The tree structure is as follows:

- <!DOCTYPE html>
- <html lang="en">
- ▶ <head>...</head>
- ▼ <body>
- .. <div></div> == \$0
- <script type="text/javascript">
- \$('p').remove();
- </script>
- </body>
- </html>

The selected element is the <div> element, which is highlighted in blue. The console shows the execution of the script, which removes all <p> elements from the DOM.

jQuery обекти и събития

jQuery – обекти и събития

Селектираните с jQuery DOM елементи не са обикновени DOM обекти. Те имат повече свойства и методи в сравнение с JS DOM обекти.

- `addClass()`, `removeClass()`, `toggleClass()`
- `on(event, callback)` за закачане на събития
- `animate()`, `fadeIn()`, etc...

jQuery – обекти и събития

Методи за променяне на елементите

`jQuery.css("color", "#f3f")`

`jQuery.html()` връща `innerHTML`

`jQuery.html(content)` – дава стойност на `innerHTML`

`jQuery.text(content)` – задава `innerHTML`, като ескейпва съдържанието

jQuery – обекти и събития

jQuery притежава удобни методи за добавяне и премахване на събития

on() и **off()**

```
function onButtonClick(){  
    $(".selected").removeClass("selected");  
    $(this).addClass("selected");  
}
```

```
$("a.button").on("click", onButtonClick);
```

jQuery – обекти и събития

Two common jQuery event methods

Event method	Description
<code>ready(handler)</code>	The event handler runs when the DOM is ready.
<code>click(handler)</code>	The event handler runs when the selected element is clicked.

Two ways to code an event handler for the jQuery ready event

The long way

```
$(document).ready(function() {  
    alert("The DOM is ready");  
});
```

The short way

```
$(function(){  
    alert("The DOM is ready");  
});           // (document).ready is assumed
```

jQuery – обекти и събития

За демо

An event handler for the click event of all h2 elements

```
$("h2").click(function() {  
    alert("This heading has been clicked");  
});
```

The click event handler within the ready event handler

```
$(document).ready(function() {  
    $("h2").click(function() {  
        alert("This heading has been clicked");  
    });  
}); // end of click event handler  
// end of ready event handler
```

A summary of the most useful jQuery methods

Method	Description
<code>next([selector])</code>	Get the next sibling of each selected element or the first sibling of a specified type if the parameter is coded.
<code>prev([selector])</code>	Get the previous sibling of each selected element or the previous sibling of a specified type if the parameter is coded.
<code>attr(attributeName)</code>	Get the value of the specified attribute from the first selected element.
<code>attr(attributeName, value)</code>	Set the value of the specified attribute for each selected element.
<code>css(propertyName)</code>	Get the value of the specified property from the first selected element.
<code>css(propertyName, value)</code>	Set the value of the specified property for each selected element.
<code>addClass(className)</code>	Add one or more classes to the selected elements and, if necessary, create the class. If you use more than one class as the parameter, separate them with spaces.
<code>removeClass([className])</code>	Remove one or more classes. If you use more than one class as the parameter, separate them with spaces.
<code>toggleClass(className)</code>	If the class is present, remove it. Otherwise, add it.
<code>hide([duration])</code>	Hide the selected elements. The duration parameter can be “slow”, “fast”, or a number giving the time in milliseconds. By default, the duration is 400 milliseconds, “slow” is 600 milliseconds, and “fast” is 200 milliseconds.
<code>show([duration])</code>	Show the selected elements. The duration parameter is the same as for the hide method.
<code>each(function)</code>	Run the function for each element in an array.

A summary of the most useful jQuery event methods

Event method	Description
<code>ready(handler)</code>	The handler runs when the DOM is ready.
<code>unload(handler)</code>	The handler runs when the user closes the browser window.
<code>error(handler)</code>	The handler runs when a JavaScript error occurs.
<code>click(handler)</code>	The handler runs when the selected element is clicked.
<code>dblclick(handler)</code>	The handler runs when the selected element is double-clicked.
<code>mouseenter(handler)</code>	The handler runs when the mouse pointer enters the selected element.
<code>mouseover(handler)</code>	The handler runs when the mouse pointer moves over the selected element.
<code>mouseout(handler)</code>	The handler runs when the mouse pointer moves out of the selected element.
<code>hover(handlerIn, handlerOut)</code>	The first event handler runs when the mouse pointer moves into an element. The second event handler runs when the mouse pointer moves out.
<code>event.preventDefault()</code>	Stops the default action of an event from happening.

jQuery – обекти и събития

Оптимизиране на събития – добавяне на родителски елемент

```
function onListItemClick(){  
    $(".selected").removeClass("selected");  
    $(this).addClass("selected");  
}  
  
$("ul").on("click", "li", onListItemClick);
```

jQuery – обекти и събития

Chaining

jQuery методите винаги връщат – резултат или **this**, което ни дава възможност да ги прикачваме последователно – като верига един за друг –

```
$('<button>')  
  .addClass('btn-success')  
  .html('Click me for success')  
  .on('click', onSuccessButtonClick)  
  .appendTo(document.body);
```

`$('<button>')` – създава бутон

`$('button')` – селектира бутон