

# jQuery

"write less, do more"

## Какво е jQuery?

jQuery е нищо друго освен един файл написан на JavaScript с функции и се използва като всеки друг файл написан на JavaScript. Вмъква се в дадена страница чрез тага `<script>` сложен в `<head>` , най-често, или някаде в `<body>`

Ако имаме файл `index.html` в папка `html`, в папката имаме друга папка `js` и в нея файл с име `jquery.js` . Зада използваме възможностите на `jquery` трябва да го включим по следния начин

```
<script type="text/javascript" src="js/jquery.js"></script>
```

Не е сложно нали, все едно си вмъкваме всеки друг `js` файл.

jQuery е доста голям JavaScript файл и може да не искате да го използвате целия, затова има две версии-едната е с минимума, който трябва да а другата е пълната версия. jQuery се обновява , оптимизира и се допълва постоянно , много хора решават да пишат отделни библиотеки базирани на jQuery , и всъщност това е най-големият плюс за да го използваме.

## Как се използва jQuery?

Представете си, че искате когато натиснете един линк ( `<a href="...">` ) и да излиза съобщение, как бихте го направили само с JavaScript. Най-вероятно така:

```
<a href="javascript:;" onclick="alert('Hello World')" >click</a>
```

Добре, става работи, когато кликнем на линка не ни праща на никаде, извежда ни съобщението и всичко е ок, нали? Но така кода става по дълъг ако трябва за всеки линк по отделно да пишем това, ако викаме функция при `onclick` лесно потребителя може да промени функцията и да извика своя собствена, написана в адресната линия. Не е особено фатално, но ако може и да не искате да го прави. Ето как можете да затрудните някой лош потребител и да имате същата функционалност, но лесно да бъде разширена и да е по добре организирана

```
<html>  
<head>  
<script type="text/javascript"  
  
src="http://ajax.googleapis.com/ajax/libs/jquery/1.4/jquery.min.js"></script>  
  
<script type="text/javascript" >
```

```

$(document).ready( function() {

    $("a.jquerylinks").click(function(event) {
        alert('hello jQuery');
        event.preventDefault();
    });

});

</script>
</head>

<body>

    <a href="javascript:;" onclick="alert('Hello World')" >click</a>
    </br>

    <a href="" class="jquerylinks">jq link</a>
    <a href="" class="jquerylinks">jq link</a>
    <a href="" class="jquerylinks">jq link</a>

</body>

</html>

```

## Селектори

В горния пример се използват класове за да се обърнем към линковете който искаме когато бъдат натискани да се покаже съобщението „hello jQuery”. Можем да кажем, че искаме всички линкове, или само един определен, когато бъдат натиснати да се показва това съобщение.

### Чрез клас и id

`$("*")` - избира всички елементи

`$("a")` - за всички тагова `<a>`

`$("a.link")` - за всички тагове `<a>` с клас „link”

`$("a#mylink")` - за всички такова `<a>` с `id="mylink"`

или

`$(".link")` - за всички тагове с клас „link”

`$("#mylink")` - за таг с `id="mylink"`

`$("#mylink.link")` - таг с `id="mylink"` и клас „link”

`$(".link.home")` - за всички тагове с класове (едновременно) „link” и „home”

### По име

Ако имаме елемент `<p name="jquery"> jquery.com </p>`

можем да се обърнем към него с `$("p[name='jquery']")`

`$("tagname[name = 'value' ]")`

ако имаме `<input name="inpts[]" type="text" />` ще трябва да използваме `'\'` за

да се обърнем правилно към елементите `$("#input[name=inpts\\[\\]] ")`

*още полезни начини да изберете елемент по неговото име са*

`selector[name]='string'` – всички които запозват или са равни на дадения  
'string'  
`selector[name]*='string'` – всички които има име което съдържа 'string' като  
substring  
`selector[name]$='string'` – всички чието име завършва на със 'string'

**\*това важи за всички атрибути не само за атрибута name**

Елементи могат да се избират и по тяхната последователност, ако имаме един списък с продукти (`<ul> <li>Портокали</li> </ul>` ) и искаме да изберем само първия продукт, само последния (незнаем колко продукта има), или някой продукт по неговият пореден номер можем да се обърнем към тях по следния начин:

`$("#ul li:first")` - първия продукт  
`$("#ul li:last")` - последния продукт  
`$("#:first-child")` - всички елементи който са първи  
`$("#:nth-child( n )")` - n-тия елемент  
пример за `nth-child()`:

`$("#ul.products li:nth-child(4)").html("Ягоди");`

Това би задало на нашия списък с продукти , четвъртият да бъде ягоди.

`$("# :odd")` - избира всички елементи с нечетен номер, `even` избира всички с четен номер

`$("# :not")` - избира всички елементи освен тези

Ако искаме да разберем поредния номер на даден елемент можем да го вземем чрез метода `index()`, който има като параметър елемента чийто номер искаме да разберем и се извиква за елемента спрямо който искаме да разберем номера му

пример:

```
<ul class="cats">
  <li>cat 1</li>
  <li>cat 2</li>
  <li class="selected">cat 3</li>
  <li>cat 4</li>
</ul>
```

за да разберем на кое място е елемента с клас `selected` трябва да се го вземем така:

```
var cat=$("#li.selected").index("#ul.cats li");
```

**!Броенето по индекс започва от 0, докато броенето при `:nth-child()` започва от 1.**

## Какво е `$(document).ready()` ?

Когато искаме да изпълваме код на jQuery, ние трябва да сме сигурни че страницата е заредена. Методът `ready()` приема като параметър функция, която всъщност е това което се изпълнява

### пример:

```
$(document).ready( function () { alert("Hello jQuery"); });
```

или

```
$(document).ready( start() );  
function start() { alert("Hello jQuery"); }
```

Не винаги е задължително да пишете вашият код в `$(document).ready()` . Можете да използвате възможностите на jQuery както всички останали js функции без да ги слагате в `ready()`, но е препорачително. Можете да имате дори няколко пъти това извикване т.е **`$(document).ready( function () { alert("Hello jQuery"); });`** може да се повтаря няколко пъти.

Както споменахме има случаи когато може да не се изпълни методът `ready()` първо. Следният пример , колкото и да е невероятно да направите точно това в ваша страница , показва ситуация която бихте предпочели да сложите вашият код извън `$(document).ready()` :

```
<html>  
<head>  
<script type="text/javascript"  
src="http://ajax.googleapis.com/ajax/libs/jquery/1.4/jquery.min.js"></script>  
<script type="text/javascript">  
  
    $(document).ready(function() {  
        alert('!inside document.ready() 3');  
    });  
  
    alert('1');  
</script>  
  
</head>  
    <body onload="alert('LAST')">  
        <script type="text/javascript">  
            alert("2");  
        </script>  
    </body>  
</html>
```

Ако ви се наложи за някой елемент да имате jquery функции, който да го променят, и сте забелязали, че те излизат първо нормално и промените се отразяват след някакъв малък интервал от време, може да е дори част от секундата, но да се забелязва и да дразни, най-доброто място е да вмъкнете `<script te='text/javascript'></script>` веднага след елемента, така ще сте сигурни, че той ще се зареди и промените ще станат

веднага след зареждането му. В този случай не се пише `$(document).ready()`.

## JqTransform

Това е най-точно казано пакет, базиран на jQuery. Представлява набор от картинки, CSS и JavaScript. Предназначението му е да направи вашия сайт по привлекателен и да избегнете някои ограничения за дизайн контролите във формулярите.

Как работи? Включвате необходимите файлове `jqtransform.js`, `jqtransform.css` и `jquery.js` и извиквате функцията `jqTransform()` за формуляра който искате да има по хубав вид. Той е opensource и е желателно да го промените в зависимост от вашите нужди.

Пример:

```
<html>
  <head>
    <link href="jqtransform/jqtransform.css" rel="stylesheet" type="text/css" />
    <script src="jqtransform/jquery.js" type="text/javascript" ></script>
    <script src="jqtransform/jqtransform.js" type="text/javascript">
  </script>
  <script type="text/javascript">
    $(document).ready( function() {
      alert("saasda");
      $("form.funky").jqTransform();
    });
  </script>
  </head>
  <body>
    <form class="funky">
      <fieldset>
        <input type="radio" > <label>1</label></br>
        <input type="radio" ><label>2</label></br>
        <input type="checkbox" ><label>3</label></br>
        <input type="checkbox" ><label>4</label></br>
      </fieldset>
      <fieldset>
        <p>Text area:</p>
        <textarea row="30" cols="40" >
        </textarea>
      </fieldset>
      <fieldset>
        <select>
          <option>1</option>
          <option>2</option>
          <option>3</option></select>
    </form>
  </body>
</html>
```

```
<input type="text" />
<input type="text" />
<input type="text" />
</fieldset>
<input type="submit" />
</form> <br/></body></html>
```

## PrettyPhoto

PrettyPhoto е полезен плъгин за разкрасяване, използва се много лесно и за почти всичко, за картици, за галерия от картинки, видео, флаш, youtube content , обикновен текст, QuickTime .

Използва се като jqTransform, добавят се

```
<link href="prettyPhoto\css\prettyPhoto.css" rel="stylesheet" type="text/css" />
<script src="prettyPhoto\js\jquery.js" type="text/javascript"> </script>
<script src="prettyPhoto\js\jquery.prettyPhoto.js" type="text/javascript"> </script>
<script type="text/javascript">
    $(document).ready(function(){
        $("a[rel^='prettyPhoto']").prettyPhoto();
    });
</script>
```

и

```
<a href="img_path" rel="prettyPhoto">  </a>
```

За галерия от снимки се използва функцията

```
$.prettyPhoto.open(pics_array,title_array,desc_array);
```

извикана при събитието **click()** .

*\*Ефектите с jQuery са много, но се използват по един и същи лесен начин.*

## JavaScript injections

Да напишем в адресното поле следния текст:

```
javascript: alert ('Hello JavaScript');
```

,това ще ни изведе съобщението „Hello JavaScript“ на която и страница да се намираме, ето един по-полезен и обасен пример.

```
Javascript:alert(document.cookie);
```

,така много лесно можем да разберем какви са бисквитките за съответната страница/сайт.

Нека да имаме дефинирана функция на js :

```
function test() { for(i=0;i<10;i++) document.write(i); }
```

Лесно можем да да сменим самата функция като напишем в адресното полес следния пример код:

```
javascript:function test() { alert("this function is changed"); }
```

*пример:*

<http://vicove.com/> имат система за гласуване, която се базира на взимане на стойност от радио бутона и изчисляването на рейтинг спрямо нея. Как можем да се възползваме от това?!

Избираме си картинка, виц или нещо друго за което може да се гласува и пишем в адрес поле:

[javascript:alert\(document.getElementsByName\('...'\)\[???\].value=value\);](#)

и после се натиска бутона 'гласувай'. Дано скоро да разберат грешката си и да я оправят.