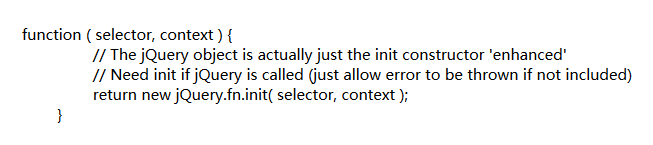
jQuery是一个JavaScript脚本库，不需要特别的安装，只需要我们在页面 <head> 标签内中，通过 script 标签引入 jQuery 库即可

<script type="text/javascript" src="http://libs.baidu.com/jquery/1.9.1/jquery.js"></script>（此版本）

<body>中输入<body>

<script type="text/javascript"> alert($) </script>

当弹出



即为环境搭建成功

1 <script src="http://libs.baidu.com/jquery/1.9.1/jquery.min.js"></script>

<script type="text/javascript">

$(document).ready(function() {

$("div").html("您好！通过慕课网学习jQuery才是最佳的途径。");

});

</script>

$(document).ready 的作用是等页面的文档（document）中的节点都加载完毕后，再执行后续的代码，因为我们在执行代码的时候，可能会依赖页面的某一个元素，我们要确保这个元素真正的的被加载完毕后才能正确的使用

**2 jQuery对象和Dom对象（两者是不同的）**

http://www.imooc.com/code/8108

<p id=”imooc”></p>

**普通处理javascript**

<script type="text/javascript">

window.onload = function(){

// 通过原生JS语法获取id为imooc1的元素p

var p = document.getElementById('imooc1');

// 将元素p在html中内容改变

p.innerHTML = 'P1：您好！通过慕课网学习jQuery才是最佳的途径';

// 将元素p的内容颜色改为红色

p.style.color = 'red';

}

</script>通过原生DOM模型提供的document.getElementById(“imooc”) 方法获取的DOM元素就是一个DOM对象，再通过innerHTML与style属性处理文本与颜色。

**jQuery处理**

<script type="text/javascript">

$(document).ready(function() {

/\*\*

\* 通过jQuery语法获取id为imooc2的元素获得一个jQuery对象

\* 调用该对象的html()方法进行更改内容

\* 调用该对象的css()方法进行更改颜色样式

\*/

var $p = $('#imooc2');

$p.html('P2：您好！通过慕课网学习jQuery才是最佳的途径').css('color','red');

});

</script>通过$('#imooc')方法会得到一个$p的jQuery对象，$p是一个类数组对象。这个对象里面包含了DOM对象的信息，然后封装了很多操作方法，调用自己的方法html与css，得到的效果与标准的JavaScript处理结果是一致的。

通过标准的JavaScript操作DOM与jQuyer操作DOM的对比，我们不难发现：

通过jQuery方法包装后的对象，是一个类数组对象。它与DOM对象完全不同，唯一相似的是它们都能操作DOM。

通过jQuery处理DOM的操作，可以让开发者更专注业务逻辑的开发，而不需要我们具体知道哪个DOM节点有那些方法，也不需要关心不同浏览器的兼容性问题，我们通过jQuery提供的API进行开发，代码也会更加精短。

**jQuery对象转化成DOM对象**

var $div = $('div') //jQuery对象

var div = $div[0] //转化成DOM对象

div.style.color = 'red' //操作dom对象的属性

**通过jquery再带的get()**

var $div = $('div') //jQuery对象

var div = $div.get(0) //通过get方法，转化成DOM对象

div.style.color = 'red' //操作dom对象的属性

**DOM对象转化成jQuery对象**

如果传递给$(DOM)函数的参数是一个DOM对象，jQuery方法会把这个DOM对象给包装成一个新的jQuery对象

通过$(dom)方法将普通的dom对象加工成jQuery对象之后，我们就可以调用jQuery的方法了

var div = document.getElementsByTagName('div'); //dom对象

var $div = $(div); //jQuery对象

var $first = $div.first(); //找到第一个div元素

$first.css('color', 'red'); //给第一个元素设置颜色

**jQuery选择器之id选择器**

id选择器：一个用来查找的ID，即元素的id属性，jQuery内部使用JavaScript函数document.getElementById()来处理ID的获取

$( "#id" )

**之类选择器**

jQuery使用JavaScript的原生getElementsByClassName()函数来实现的

$( ".class" )

<div class="aaron">

<p>class="aaron"</p>

<p>选中</p>

</div>

<script type="text/javascript">

//通过原生方法处理

//样式是可以多选的，所以得到的是一个合集

//需要通过循环给合集中每一个元素修改样式

var divs = document.getElementsByClassName('aaron');

for (var i = 0; i < divs.length; i++) {

divs[i].style.border = "3px solid blue";

}

</script>

<script type="text/javascript">

//通过jQuery直接传入class

//class选择器可以选择多个元素

$(".aaron")?.css("border", "3px solid red");

</script>

不难想到$(".imooc").css()方法内部肯定是带了一个隐式的循环处理，

**之元素选择器**

通过getElementsByTagName方法得到页面所有的<div>元素

var divs = document.getElementsByTagName('div');

同样的效果，$("p")选取所有的<p>元素，通过css方法直接赋予样式

$("p")

**之全选择器**

在CSS中，经常会在第一行写下这样一段样式

\* {padding: 0; margin: 0;}

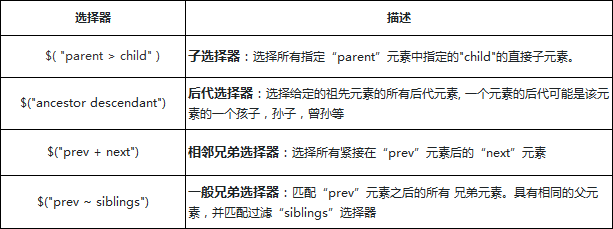
**通配符\***意味着给所有的元素设置默认的边距。jQuery中我们也可以通过传递\*选择器来选中文档页面中的元素

描述：

$( "\*" )

通过document.getElementsByTagName()中传递"\*"同样可以获取到

**jQuery选择器之层级选择器/\* 存在问题\*/**



$(".prev ~ div")?.css("border", "3px groove blue"); 具体样式的写法