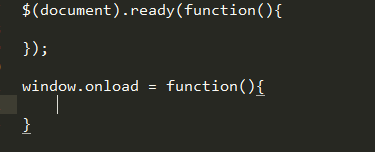
1.2

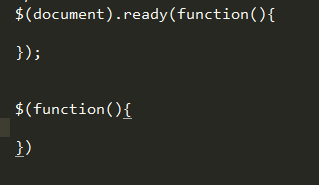


上述两段代码在功能上可以交换，但他们又有许多区别：

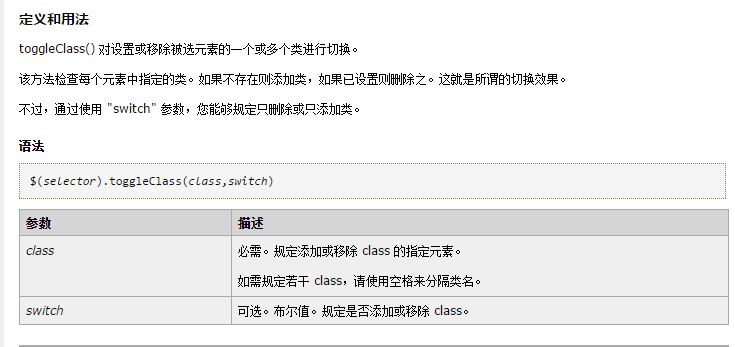
执行时间不同：$(document).ready在页面框架下载完毕后就执行；而window.onload必须在页面全部加载完毕（包括图片下载）后才执行。很明显前者的执行效率高于后者。

执行数量不同：$(document).ready可以重复写多个，并且每次执行结果不同；而window.onload尽管可以执行多个，但仅输出最后一个执行结果，无法完成多个结果的输出。

$(document).ready(function(){})可以简写成$(function(){}).因此下面的代码是等价的。



1-4：jQuery的addClass()方法，toggleClass(className)方法



2-1

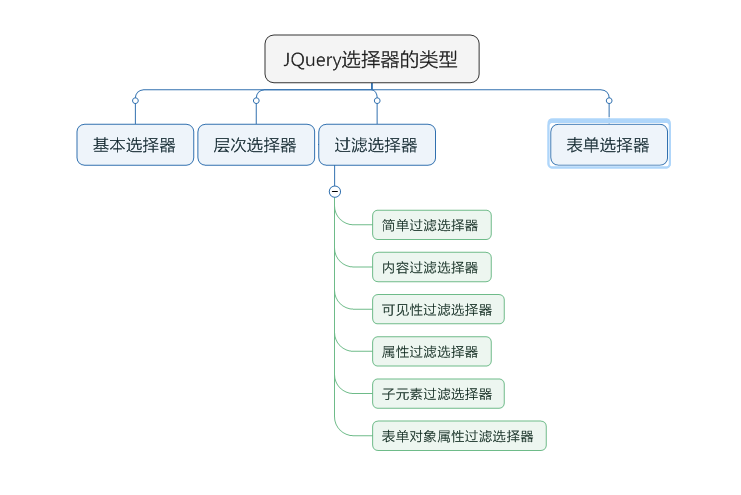
Cellpadding:

表格单元边界与单元内容之间的间距

Cellspacing：

表格单元格间距

jQuery的:nth-child(n)选择器用于匹配作为父元素下的第n个（或特定顺序的）子元素的元素，将其封装成jQuery对象并返回。



2.2.3 简单过滤选择器

根据某类过滤规则进行元素的匹配，书写时都以冒号（：）开头

|  |  |  |
| --- | --- | --- |
| 选择器 | 功能描述 | 返回值 |
| First() :first | 获取第一个元素 | 单个元素 |
| Last() :last | 获取最后一个元素 | 单个元素 |
| :not(selector) | 获取除给定选择器外的所有元素 | 元素集合 |
| :even | 获取所有索引值为偶数的元素，索引号从0开始 | 元素集合 |
| :odd | 获取所有索引值为基数的元素，索引号从0开始 | 元素集合 |
| :eq(index) | 获取指定索引值的元素，从0开始 | 单个元素 |
| :gt(index) | 获取所有大于给定索引值的元素，从0开始 | 元素集合 |
| :lt(index) | 获取所有小于给定索引值的元素，从0开始 | 元素集合 |
| :header | 获取所有标题类型的元素，如：h1,h2 | 元素集合 |
| .animated | 获取正在执行动画效果的元素 | 元素集合 |

2.2.4 内容过滤选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 功能描述 | 返回值 |
| :contains(text) | 获取包含给定文本的元素 | 元素集合 |
| :empty | 获取所有不包含子元素或者文本的空元素 | 元素集合 |
| :has(selector) | 获取含有选择器所匹配的元素 | 元素集合 |
| :parent | 获取含有子元素或者文本的元素 | 元素集合 |

2.2.5 可见性过滤选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 功能描述 | 返回值 |
| :hidden | 获取所有不可见元素。或者type为hidden的元素 | 元素集合 |
| :visible | 获取所有的可见元素 | 元素集合 |

2.2.6 属性过滤选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 功能描述 | 返回值 |
| [attribute] | 获取包含给定属性的元素 | 元素集合 |
| [attribute=value] | 获取等于给定的属性是某个特定值的元素 | 元素集合 |
| [attribute!=value] | 获取不等于给定的属性是某个特定值的元素 | 元素集合 |
| [attribute]^=value | 获取给定的属性是以某些值开始的元素 | 元素集合 |
| [attribute]$=value | 获取给定的属性是以某些值结尾的元素 | 元素集合 |
| [attribute]\*=value | 获取给定的属性是以包含某些值的元素 | 元素集合 |
| [selector1][selector2][selector3] | 获取满足多个条件的复合属性的元素 | 元素集合 |

2.2.7 子元素过滤选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 功能描述 | 返回值 |
| :nth-child(eq|even|odd|index) | 获取每个父元素下的特定位置元素，索引号从1开始 | 元素集合 |
| :first-child | 获取每个父元素下的第一个子元素 | 元素集合 |
| :last-child | 获取每个父元素下的最后一个子元素 | 元素集合 |
| :only-child | 获取每个父元素下的仅有一个子元素 | 元素集合 |

Css ul list-style-type:none;

2.2.8表单对象属性过滤选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 功能描述 | 返回值 |
| :enabled | 获取表单中所有属性为可用的元素 | 元素集合 |
| :disabled | 获取表单中所有属性为不可用的元素 | 元素集合 |
| :checked | 获取表单中所有被选中的元素 | 元素集合 |
| :selected | 获取表单中所有被选中option的元素 | 元素集合 |

2.2.9 表单选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 功能描述 | 返回值 |
| :input | 获取所有input，textarea，select | 元素集合 |
| :text | 获取所有单行文本框 | 元素集合 |
| :password | 获取所有密码框 | 元素集合 |
| :radio | 获取所有单选按钮 | 元素集合 |
| :checkbox | 获取所有复选框 | 元素集合 |
| :submit | 获取所有提交按钮 | 元素集合 |
| :image | 获取所有图像域 | 元素集合 |
| :reset | 获取所有重置按钮 | 元素集合 |
| :button | 获取所有按钮 | 元素集合 |
| :file | 获取所有文件域 | 元素集合 |

3.2.1 获取元素的属性：

Attr(name)

3.2.2 设置元素的属性

Attr(key,value);

Attr( {key0:value0,key1:value1 } )

Attr()方法还可以绑定一个function（）函数，通过该函数返回的值作为元素的属性值，语法： attr（key,function(index)）,其中，参数index为当前元素的索引号，整个函数返回一个字符串作为元素的属性值。

3.2.3 删除元素的属性

removeAttr(name)

3.3.1 获取和设置元素

在jQuery中，操作元素内容的方法包括html()和text()

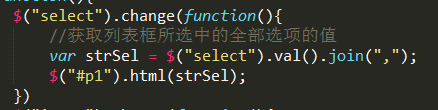
|  |  |  |
| --- | --- | --- |
| 语法格式 | 参数说明 | 功能描述 |
| Html() | 无参数 | 用于获取元素的html内容 |
| Html(val) | Val参数为元素的html内容 | 用于设置元素的html内容 |
| Text() | 无参数 | 用于获取元素的文本内容 |
| Text(val) | Val参数为元素的文本内容 | 用于设置元素的文本内容 |

3.3.2 获取和设置元素的值

Val(val)

如果不带参数val，是获取某元素的值；反之，则是将参数val的值赋给某元素，即设置元素的值。常用于表单中获取和设置对象的值。

通过val（）方法还可以获取select标记中的多个选项值：val().join(“,”)



3.4 元素样式操作

1. 直接设置元素样式值：css(name,value)
2. 增加元素css类别：addClass(class) 多个：addClass(class0 class1 …)
3. 切换元素css样式：通过toggleClass（）方法切换不同的元素类别

toggleClass(class):其功能是当元素中含有名称为class的css类别时，删除该类别，否则增加一个该名称的css类别。

1. 删除元素css类别： removeClass([class]).

$(“p”).removeClass(); //删除p标记的全部类别。

3.5 页面元素操作

3.5.1创建元素节点

函数$() 用于动态创建页面元素

3.5.2 内部插入节点

|  |  |  |
| --- | --- | --- |
| 语法格式 | 参数说明 | 功能描述 |
| Append(content) | Content表示追加到目标中的内容 | 像所选择的元素内部插入内容 |
| Append(function(index,html)) | 通过function函数返回追加到目标中的内容 | 像所选择的元素内部插入function函数所返回的内容 |
| AppendTo(content) | Content表示被追加的内容 | 把所选择的元素追加到另一个指定的元素集合中 |
| Prepend(content) | Content表示插入目标元素内部前面的内容 | 向每个所选择的元素内部前置内容 |
| Prepend(function(index,html)) | 通过function函数返回插入目标元素内部前面的内容 | 向所选择的元素内部前置function函数所返回的内容 |
| prependTo(content) | Content表示用于选择元素的jQuery表达式 | 将所选择的元素前置到另一个指定的元素集合中 |

Append（function（index，html））：将一个function函数作为append方法的参数，函数的功能必须返回一个字符串，作为append方法插入的内容。其中index参数为对象在这个集合中的索引值，html参数为该对象原有的html值。3-11.html

3.5.3 外部插入节点

|  |  |  |
| --- | --- | --- |
| 语法格式 | 参数说明 | 功能描述 |
| After(content) | Content表示插入目标元素外部后面的内容 | 向所选择的元素外部后面插入内容 |
| After(function) | 通过function函数返回插入目标外部后面的内容 | 向所选择的元素外部后面插入function函数所返回的内容 |
| Before(content) | 通过function函数返回插入目标外部前面的内容 | 向所选择的元素外部前面插入内容 |
| Before(function) | 通过function函数返回插入目标外部前面的内容 | 向所选择的元素外部前面插入function函数所返回的内容 |
| insertAfter(content) | Content表示插入目标元素外部后面的内容 | 将所选择的元素插入另一个指定的元素外部后面 |
| insertBefore(content) | Content表示插入目标元素外部前面的内容 | 将所选择的元素插入另一个指定的元素外部前面 |

3.5.4 复制元素节点

Clone():功能为复制匹配的DOM元素并且选中复制成功的元素，该方法仅是复制元素本身，被复制后的新元素不具有任何元素行为。如果需要在复制时将该元素的全部行为也进行复制，可以通过方法clone（true）实现。

3.5.5 替换元素节点 replaceWith() replaceAll()

replaceWith(content):将所有选择的元素替换成指定的html或dom元素，其中参数content为被所选择元素替换的内容.

replaceAll(selector)：将所有选择的元素替换成指定selector的元素，其中参数selector为需要被替换的元素。



3.5.6 包裹元素节点

|  |  |  |
| --- | --- | --- |
| 语法格式 | 参数说明 | 功能描述 |
| Wrap(html) | Html参数为字符串代码，用于生成元素并包裹所选元素 | 把所有选择的元素用其他字符串代码包裹起来 |
| Wrap(elem) | Elem参数用于包装所选元素的dom元素 | 把所有选择的元素用其他dom元素包装起来 |
| Wrap(fn) | Fn参数为包裹结构的一个函数 | 把所有选择的元素用function函数返回的代码包装起来 |
| Unwrap() | 无参数 | 移除所选元素的父元素或包裹标记 |
| WrapAll(html) | Html参数为字符创代码，用于生成元素并包裹所选元素 | 把所有选择的元素用单个元素包裹起来 |
| wrapAll(elem) | Elem参数用于包装所选元素的dom元素 | 把所有选择的元素用单个dom元素包裹起来 |
| wrapInner(html) | Html参数为字符串代码，用于生成元素并包裹所选元素 | 把所有选择的元素的子内容（包括文本节点）用字符串代码包裹起来 |
| wrapInner(elem) | Elem参数用于包装所选元素的dom元素 | 把所有选择的元素的子内容（包括文本节点）用dom元素包裹起来 |
| wrapInner(fn) | Fn参数为包裹结构的一个函数 | 把所有选择的元素的子内容（包括文本节点）用function函数返回的代码包裹起来 |

3.5.7 遍历元素

Each(callback)

参数callback是一个function函数，该函数还可以接受一个形参index，此形参为遍历元素的序号（从0开始），如果需要访问元素中的属性，可以借助形参index，配合this关键字来实现元素属性的设置或获取。

3.5.8 删除页面元素

Remove（） 和empty（）

使用empty（）仅仅可以清空全部的节点或节点所包括的所有后代元素，并非删除节点和元素。

Remove([expr])

第4章 jQuery中的事件与应用

4.1.1 事件中的冒泡现象

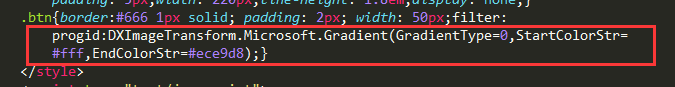
严格来说，事件在触发后分为两个阶段，一个是捕获（capture），另一个则是冒泡（Bubbling）；但有些遗憾的是，大多数浏览器并不是都支持捕获阶段，jQuery也不支持。因此在事件触发后，往往执行冒泡过程。所谓的冒泡其实实质就是事件执行中的顺序。

* + 1. 如何阻止事件的冒泡

即单击按钮就执行单一的单击事件，并不触发其他外围的世界。在jQuery中，可以通过stopPropagation（）方法可以阻止冒泡过程的发生。

在代码中，除了使用stopPropagation（）方法阻止事件的冒泡过程外，还可以通过语句return false实现停止事件的冒泡过程。

知识点：



FILTER:progid:DXImageTransform.Microsoft.Gradient使用   
语法：   
filter:progid:DXImageTransform.Microsoft.Gradient(enabled=bEnabled,startColorStr=iWidth,endColorStr=iWidth)   
属性：   
enabled:可选项。布尔值(Boolean)。设置或检索滤镜是否激活。 true | false   
　　true: 默认值。滤镜激活。   
　　false:滤镜被禁止。

startColorStr:可选项。字符串(String)。设置或检索色彩渐变的开始颜色和透明度。   
　　其格式为 #AARRGGBB 。 AA 、 RR 、 GG 、 BB 为十六进制正整数。取值范围为 00 - FF 。 RR 指定红色值， GG 指定绿色值， BB 指定蓝色值，参阅 #RRGGBB 颜色单位。 AA 指定透明度。 00 是完全透明。 FF 是完全不透明。超出取值范围的值将被恢复为默认值。   
　　取值范围为 #FF000000 - #FFFFFFFF 。默认值为 #FF0000FF 。不透明蓝色。   
EndColorStr:可选项。字符串(String)。设置或检索色彩渐变的结束颜色和透明度。参阅 startColorStr 属性。默认值为 #FF000000 。不透明黑色。   
特性：   
Enabled:可读写。布尔值(Boolean)。参阅 enabled 属性。   
GradientType:可读写。整数值(Integer)。设置或检索色彩渐变的方向。1 | 0   
　　1:默认值。水平渐变。   
　　0:垂直渐变。   
StartColorStr:可读写。字符串(String)。参阅 startColorStr 属性。   
StartColor:可读写。整数值(Integer)。设置或检索色彩渐变的开始颜色。 取值范围为 0 - 4294967295 。 0 为透明。 4294967295 为不透明白色。   
EndColorStr:可读写。字符串(String)。设置或检索色彩渐变的结束颜色和透明度。参阅 startColorStr 属性。默认值为 #FF000000 。不透明黑色。   
EndColor:可读写。整数值(Integer)。设置或检索色彩渐变的结束颜色。 取值范围为 0 - 4294967295 。 0 为透明。 4294967295 为不透明白色。当在脚本中使用此特性时，也可以用十六进制格式： 0xAARRGGBB 。   
说明：   
在对象的背景和内容之间显示定制的色彩层。   
当此效果通过转变显示时，在渐变册色彩层之上的文本程序性的初始化为透明的，当色彩渐变实现后，文本颜色会以其定义的值更新。

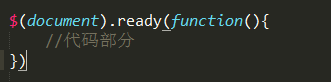
* 1. 页面载入事件

jQuery的页面载入事件ready（）方法，类似于传统JavaScript的onLoad（）方法，只不过在事件执行时间上有区别：onload（）方法的执行必须是页面中的全部元素完全加载到浏览器才触发，在这种情况下，如果页面中的图片过多或图片过大，那么有可能要等onload方法执行完毕，用户才能进行其他的操作。如果使用jQuery的ready（）方法加载页面，则只要页面的dom模型加载完毕，就会触发ready（）方法。因此，两者在事件的执行效果上ready（）方法明显优于onload（）方法。

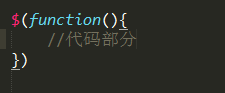
4.2.1 ready（）方法的工作原理

在jQuery脚本加载到页面时，会设置一个isReady的标记，用于监听页面加载的进度。遇到执行ready（）方法时，通过查看isReady之是否被设置，如果未被设置，那么就说明页面并未加载完成，在此情况下，将未完成的部分用一个数组缓存起来，当全部加载完成后，再将未完成的部分通过缓存一一执行。

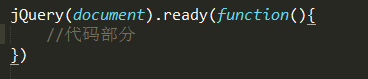
* + 1. ready（）的几种写法

一：

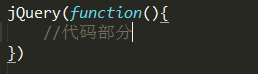
二：



三：



四：



[

<http://www.cnblogs.com/Showshare/p/different-between-attr-and-prop.html> prop and attr

4.3.1 使用bind（）方法绑定事件

Bind()功能是为每个选择元素的事件绑定处理函数，语法格式如下：

Bind(type,[data],fn)

Type为一个或多个类型的字符串，如“click”，“change”，也可以自定义类型；可以被参数type调用的类型包括blur,focus,load,resize,scroll,unload,click,dblclick,mousedown,mouseup,mousemove,mouseover,mouseout,mouseenter,mouseleave,change,select,submit,keydown,keypress,keyup,error.

参数data是作为event.data属性值传递给事件对象的额外数据对象。

参数fn是绑定到每个选择元素的事件中的处理函数。

如果要在一个元素中绑定多个事件，可以将事件用空格隔开。

4.3.2 通过映射方式绑定事件

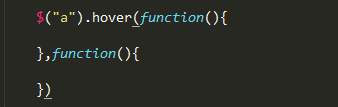
在jQuery绑定事件时，还可以通过传入一个映射，对所选对象绑定多个事件处理函数。

* 1. 切换事件

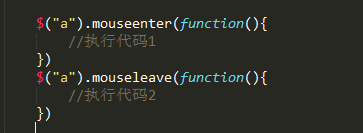
在jQuery中，有两个方法用于事件的切换，一个是hover(),另一个方法是toggle()。所谓切换事件，即有2个以上的事件绑定于一个元素，在元素的行为动作间进行切换。

4.4．1 hover（）方法

调用hover（）方法可以使元素子啊鼠标悬停与鼠标移出的事件中进行切换，也可以通过jQuery中的mouseenter与mouseleave进行替换。



等价于：



Hover()功能是当鼠标移动到所选的元素上面时，执行指定的第一个函数；当鼠标移出这个元素时，执行指定的第二个函数，其语法格式：hover(over,out)；参数over为鼠标移到元素时触发的函数，参数out为鼠标移出元素时触发的函数。