# jQuery

## jQuery介绍：

jQuery 是一个高效、精简并且功能丰富的 JavaScript 工具库。它提供的 API 易于使用且兼容众多浏览器，这让诸如 HTML 文档遍历和操作、事件处理、动画和 Ajax操作更加简单。(封装了原生JavaScript函数作为属性的一个对象)

Query 是继 Prototype 之后又一个优秀的 JavaScript 库  
jQuery 理念: 写得少, 做得多.

## jQuery的俩大特点：

### 链式编程：

比如.show()和.html()可以连写成.show().html()。

### 隐式迭代：

隐式 对应的是 显式。隐式迭代的意思是：在方法的内部进行循环遍历，而不用我们自己再进行循环，简化我们的操作，方便我们调用。

### jQuery()：

返回匹配的元素集合，无论是通过在DOM的基础上传递的参数还是创建一个HTML字符串

### 使用步骤：

（1）引包

（2）入口函数

（3）功能实现代码（事件处理）

**主要：导包的代码一定要放在js代码的最上面。**

下载jQuery包之后里面会有两个文件，一个是jquery-3.3.1.js，一个是jquery-3.3.1.min.js

它们的区别是：

* 第一个是未压缩版，第二个是压缩版。
* 平时开发过程中，可以使用任意一个版本；但是，项目上线的时候，推荐使用压缩版。

### jQuery 的入口函数和 $ 符号：

#### 入口函数：

原生 js 的入口函数指的是：**window.onload = function() {};** 如下：

//原生 js 的入口函数。页面上所有内容加载完毕，才执行。

//不仅要等文本加载完毕，而且要等图片也要加载完毕，才执行函数。

window.onload = function () {

alert(1);

}

#### jQuery的入口函数的写法：

**写法一：**

//1.文档加载完毕，图片不加载的时候，就可以执行这个函数。

$(document).ready(function () {

alert(1);

})

**写法二：**（写法一的简洁版）

//2.文档加载完毕，图片不加载的时候，就可以执行这个函数。

$(function () {

alert(1);

});

**写法三：**

//3.文档加载完毕，图片也加载完毕的时候，在执行这个函数。

$(window).ready(function () {

alert(1);

})

#### jQuery入口函数与js入口函数的区别：

**区别一：书写个数不同：**

•Js 的入口函数只能出现一次，出现多次会存在事件覆盖的问题。

•jQuery 的入口函数，可以出现任意多次，并不存在事件覆盖问题。

**区别二：执行时机不同：**

•Js的入口函数是在所有的文件资源加载完成后，才执行。这些文件资源包括：页面文档、外部的js文件、外部的css文件、图片等。

•jQuery的入口函数，是在文档加载完成后，就执行。文档加载完成指的是：DOM树加载完成后，就可以操作DOM了，不用等到所有的外部资源都加载完成。

文档加载的顺序：从上往下，边解析边执行。

### jQuery的$符号

**$ 实际上表示的是一个函数名** 如下：

$(); // 调用上面我们自定义的函数$

$(document）.ready(function(){}); // 调用入口函数

$(function(){}); // 调用入口函数

$(“#btnShow”) // 获取id属性为btnShow的元素

$(“div”) // 获取所有的div标签元素

如上方所示，jQuery 里面的 $ 函数，根据传入参数的不同，进行不同的调用，实现不同的功能。返回的是jQuery对象。

jQuery这个js库，除了$ 之外，还提供了另外一个函数：jQuery。jQuery函数跟 $ 函数的关系：jQuery === $。

## js中的DOM对象 和 jQuery对象 比较（重点，难点）

### 获取原生js的dom元素的方式不同：

通过原生 js 获取这些元素节点的方式是：

var myBox = document.getElementById("app"); //通过 id 获取单个元素

var boxArr = document.getElementsByClassName("box"); //通过 class 获取的是伪数组

var divArr = document.getElementsByTagName("div"); //通过标签获取的是伪数组 （或者通过封装函数去获取，但是都要传入参数）

通过 jQuery 获取这些元素节点的方式是：（获取的都是数组）

//获取的是数组，里面包含着原生 JS 中的DOM对象。

　　console.log($('#app'));  
　　console.log($('.box'));  
　　console.log($('div'));

**设置当前4个div的样式：**

　　　　　　$('div').css({

'width': '200px',

'height': '200px',

"background-color":'red',

'margin-top':'20px'

})

由于JQuery 自带了 css()方法，我们还可以直接在代码中给 div 设置 css 属性。

**总结**：jQuery 就是把 DOM 对象重新包装了一下，让其具有了 jQuery 方法。

### 二者的相互转换：

#### 1、 DOM 对象 转为 jQuery对象：

$(js对象);

#### 2、jQuery对象 转为 DOM 对象：

jquery对象[index]; //方式1（推荐）

jquery对象.get(index); //方式2

jQuery对象转换成了 DOM 对象之后，可以直接调用 DOM 提供的一些功能。如：

$('div')[1].style.backgroundColor = 'yellow';

$('div')[3].style.backgroundColor = 'green';

**总结**：如果想要用哪种方式设置属性或方法，必须转换成该类型。

#### 代码案例：（隔行换色）

<!DOCTYPE html>

<html>

<head lang="en">

<meta charset="UTF-8">

<title></title>

<script src="jquery-3.3.1.js"></script>

<script>

//入口函数

jQuery(function () {

var jqLi = $("li");

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

if (i % 2 === 0) {

//jquery对象，转换成了js对象

jqLi[i].style.backgroundColor = "pink";

} else {

jqLi[i].style.backgroundColor = "yellow";

}

}

});

</script>

</head>

<body>

<ul>

<li>小马哥</li>

<li>小马哥</li>

<li>小马哥</li>

<li>小马哥</li>

<li>小马哥</li>

<li>小马哥</li>

</ul>

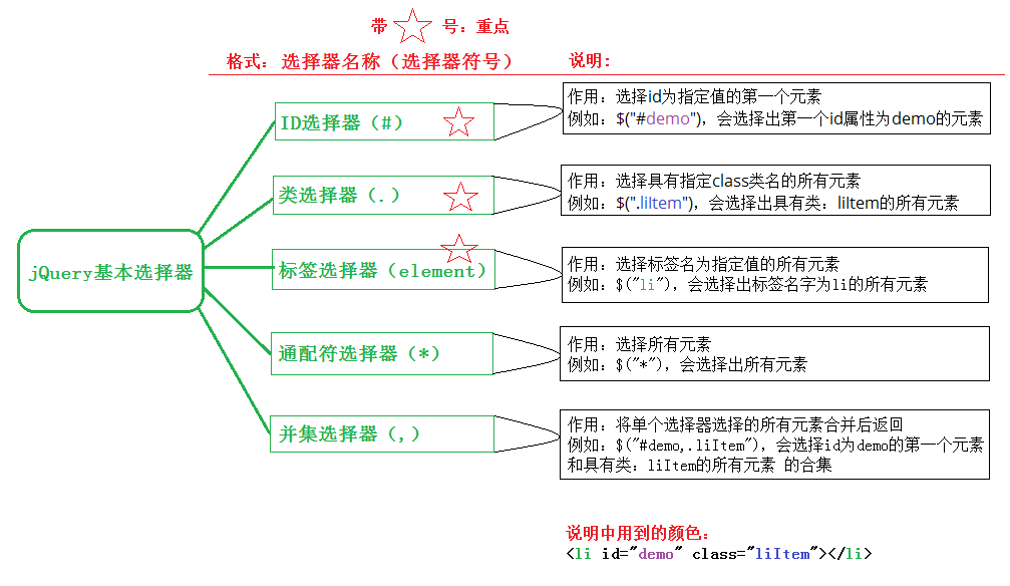
</body>

</html>

## jQuery的选择器：

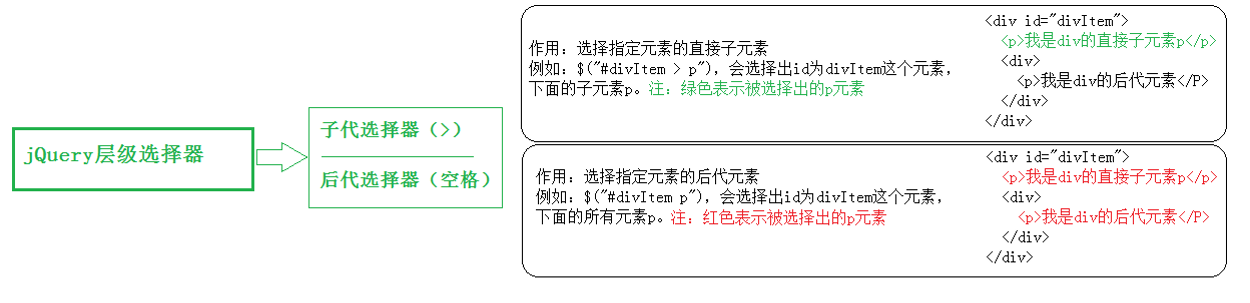
### 1、jQuery 的基本选择器





### 层级选择器：



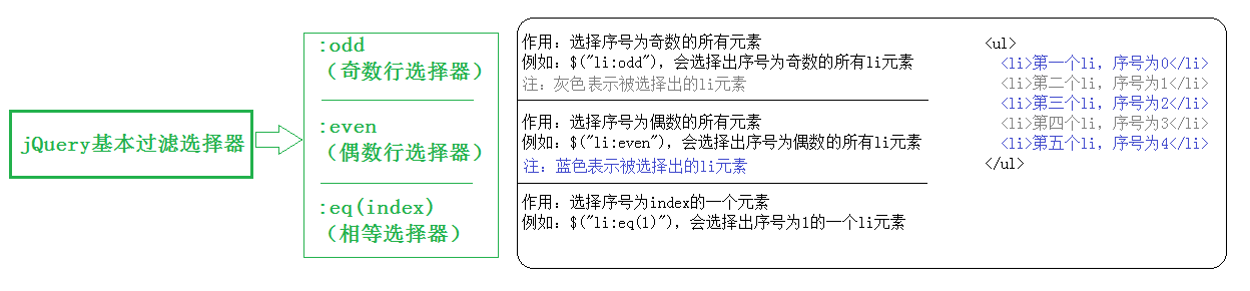


有>号的表示选择的是父层下的元素（一定是父亲的亲儿子），空格表示父层下的所有元素（不管是亲儿子还是孙子还是。。。）

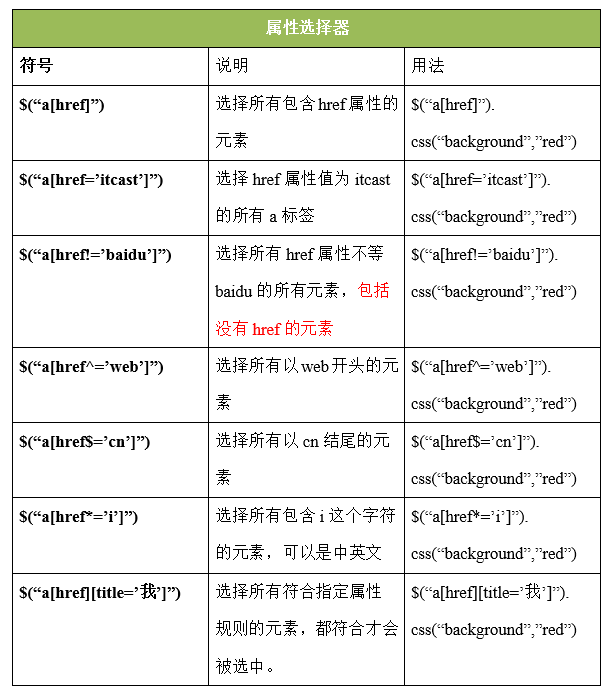
### 基本过滤选择器：



解释：



### 属性选择器：



### 筛选选择器：



#### 总结练习代码：

<!doctype html>  
<**html** lang=**"en"**>  
<**head**>  
 <**meta** charset=**"UTF-8"**>  
 <**meta** name=**"viewport"** content=**"width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0"**>  
 <**meta** http-equiv=**"X-UA-Compatible"** content=**"ie=edge"**>  
 <**title**>**choose**</**title**>  
</**head**>  
<**body**>  
 <**div** class=**"yeye"**>  
 <**div** class=**"baba"**>  
 <**ul**>  
 <**li**>**恋爱要抓紧1**</**li**>  
 <**li**>**恋爱要抓紧2**</**li**>  
 <**li** id=**"hi"**>**恋爱要抓紧3**</**li**>  
 <**li**>**恋爱要抓紧4**</**li**>  
 <**li**>**恋爱要抓紧5**</**li**>  
 <**li** id=**"name"**>  
 <**ol** class=**"ret"**>  
 <**li**>**ding**</**li**>  
 </**ol**>  
 </**li**>  
 </**ul**>  
 </**div**>  
 <**p**>  
 **丁媛媛** </**p**>  
 </**div**>  
<**ul**>  
 <**li**>**1**</**li**>  
 <**li**>**2**</**li**>  
 <**li**>**3**</**li**>  
</**ul**>  
</**body**>  
<**script** src=**"jquery-311.js"**></**script**>  
<**script**>  
 $(function(){  
 $('li').attr('type','none');  
 //所选元素属性值的修改，需要传入属性名和属性值，俩个参数  
 $('ul>li').css({  
 'width':100,  
 'height':'30px',  
 'background-color':'red',  
 'border':'1px solid black',  
 'text-align':'center',  
 'padding':10  
 });  
 $('.baba>li:odd').css(  
 'background','blue')  
 })  
</**script**>  
</**html**>

## jQuery动画效果：

### 显示动画：

#### 方式一：

$("div").show();

解释：无参数，表示让指定的元素直接显示出来。其实这个方法的底层就是通过display: block;实现的。

#### 方式二：

$('div').show(3000);

解释：通过控制元素的宽高、透明度、display属性，逐渐显示，2秒后显示完毕。

#### 方式三：

$("div").show("slow");

参数可以是：

* slow 慢：600ms
* normal 正常：400ms
* fast 快：200ms

解释：和方式二类似，也是通过控制元素的宽高、透明度、display属性，逐渐显示。

#### 方式四：

//show(毫秒值，回调函数;

$("div").show(5000,function () {

alert("动画执行完毕！");

});

解释：动画执行完后，立即执行回调函数。

**总结：**

**上面的四种方式几乎一致：参数可以有两个，第一个是动画的执行时长，第二个是动画结束后执行的回调函数。**

### 隐藏动画：

方式参照上面的show()方法的方式

$(selector).hide();

$(selector).hide(1000);

$(selector).hide("slow");

$(selector).hide(1000, function(){});

### 实现点击按钮显示盒子，再点击按钮隐藏盒子：

<!doctype html>  
<**html** lang=**"en"**>  
<**head**>  
 <**meta** charset=**"UTF-8"**>  
 <**meta** name=**"viewport"** content=**"width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0"**>  
 <**meta** http-equiv=**"X-UA-Compatible"** content=**"ie=edge"**>  
 <**title**>**Document**</**title**>  
 <**style**>  
 .fa{*width*:120**px**;*height*:300**px**;  
 *background-color*: #e175ff;  
 *display*:**none**;}  
 </**style**>  
</**head**>  
<**body**>  
 <**button** class=**"an"** id=**"an"**>**显示**</**button**>  
 <**div** class=**"fa"** id=**"fa"**></**div**>  
</**body**>  
<**script** src=**"jquery-311.js"**></**script**>  
<**script**>  
 var *ret* = true;  
 *console*.log($('div'));  
 *console*.log($('#an'));  
 //w.fn.init [div#fa.fa, prevObject: w.fn.init(1)]0: div#fa.falength: 1prevObject: w.fn.init [document]\_\_proto\_\_: Object(0)  
// lianix.html:23 w.fn.init [button#an.an] 一个是数组，一个是目标元素  
 $('#an').click(function(){  
 if(*ret*){  
 $('#fa').show(2000,function(){  
 $('#an').text('隐藏');  
 *ret*=false;  
 *console*.log('11')  
 })  
 }else{  
 $('#fa').hide(2000,function(){  
 $('#an').text('显示');  
 *ret*=true;  
 *console*.log('22')  
 })  
 }  
 })  
</**script**>  
</**html**>

### 开关式显示隐藏动画

$('#box').toggle(3000,function(){});

显示和隐藏的来回切换采用的是toggle()方法：就是先执行show()，再执行hide()。

<!doctype html>  
<**html** lang=**"en"**>  
<**head**>  
 <**meta** charset=**"UTF-8"**>  
 <**meta** name=**"viewport"** content=**"width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0"**>  
 <**meta** http-equiv=**"X-UA-Compatible"** content=**"ie=edge"**>  
 <**title**>**Document**</**title**>  
 <**style**>  
 .fa{*width*:120**px**;*height*:300**px**;  
 *background-color*: #e175ff;  
 *display*:**none**;}  
 </**style**>  
</**head**>  
<**body**>  
 <**button** class=**"an"** id=**"an"**>**显示**</**button**>  
 <**div** class=**"fa"** id=**"fa"**></**div**>  
</**body**>  
<**script** src=**"jquery-311.js"**></**script**>  
<**script**>  
 $(function(){  
 $('#an').click(function(){  
 $('#fa').toggle(1000,function(){  
 if ($('#an').text()=='隐藏'){$('#an').text('显示')}  
 else{$('#an').text('隐藏')}  
 })  
 })  
 })  
</**script**>  
</**html**>

### 滑入和滑出：

#### 1、滑入动画效果：（类似于生活中的卷帘门）

$(selector).slideDown(speed, 回调函数);

解释：下拉动画，显示元素。

注意：省略参数或者传入不合法的字符串，那么则使用默认值：400毫秒（同样适用于fadeIn/slideDown/slideUp）

#### 2、滑出动画效果：

$(selector).slideUp(speed, 回调函数);

解释：上拉动画，隐藏元素。

#### 3、滑入滑出切换动画效果：

$(selector).slideToggle(speed, 回调函数);

### 淡入淡出的动画：

#### 1、淡入动画效果：

$(selector).fadeIn(speed, callback);

作用：让元素以淡淡的进入视线的方式展示出来。

#### 2、淡出动画效果：

$(selector).fadeOut(1000);

作用：让元素以渐渐消失的方式隐藏起来

#### 3、淡入淡出切换动画效果：

$(selector).fadeToggle('fast', callback);

作用：通过改变透明度，切换匹配元素的显示或隐藏状态。

参数的含义同show()方法。

### 自定义动画：

语法：

$(selector).animate({params}, speed, callback);

作用：执行一组CSS属性的自定义动画。

* 第一个参数表示：要执行动画的CSS属性（必选）
* 第二个参数表示：执行动画时长（可选）
* 第三个参数表示：动画执行完后，立即执行的回调函数（可选）

### 停止动画：

$(selector).stop(true, false);

**里面的两个参数，有不同的含义。**

第一个参数：

* true：后续动画不执行。
* false：后续动画会执行。

第二个参数：

* true：立即执行完成当前动画。
* false：立即停止当前动画。

PS：参数如果都不写，默认两个都是false。实际工作中，直接写stop()用的多。