1、Document

document为JS中的网页文档对象

$(document)为jQuery中的网页文档对象

* $(function(){}) 为 $(document).ready(function(){})的简写

文档就绪事件：为了防止文档在完全加载之前运行jQuery代码，即在DOM加载完成之后才可以对DOM进行操作。

优先级高于window.load，可以用来初始化操作。

* (function(global){ //闭包函数

......

})(this);

定义一个匿名函数，传入this作为参数并执行，this跟程序的上下文有关。

一般匿名函数的好处是利用闭包防止其内部的变量与外部变量重名冲突

2、event事件对象的方法

$('#sidebar-collapse').click(function(e){

//取消事件的默认动作，通知 Web 浏览器不要执行与事件关联的默认动作（如果存在这样的动作）

e.preventDefault();

//不再派发事件，终止事件在传播过程的捕获、目标处理或起泡阶段进一步传播。调用该方法后，该节点上处理该事件的处理程序将被调用，事件不再被分派到其他节点。

e.stopPropagation();}

* .preventDefault()方法

该方法将通知 Web 浏览器不要执行与事件关联的默认动作（如果存在这样的动作）。例如，如果 type 属性是 "submit"，在事件传播的任意阶段可以调用任意的事件句柄，通过调用该方法，可以阻止提交表单。注意，如果 Event 对象的 cancelable 属性是 fasle，那么就没有默认动作，或者不能阻止默认动作。无论哪种情况，调用该方法都没有作用

* .stopPropagation()方法

不再派发事件。

终止事件在传播过程的捕获、目标处理或起泡阶段进一步传播。调用该方法后，该节点上处理该事件的处理程序将被调用，事件不再被分派到其他节点。

该方法将停止事件的传播，阻止它被分派到其他 Document 节点。在事件传播的任何阶段都可以调用它。注意，虽然该方法不能阻止同一个 Document 节点上的其他事件句柄被调用，但是它可以阻止把事件分派到其他节点。

4、html中的<i> 标签显示斜体文本效果。

5、jquery中的遍历方法is

is() 根据选择器、元素或 jQuery 对象来检测匹配元素集合，如果这些元素中至少有一个元素匹配给定的参数，则返回 true。

6、$.brower获取关于浏览器的信息，在jquery1.9中已经移除，用$.support替代

8、jquery的事件方法：当调整浏览器窗口的大小时，发生 resize 事件。

resize() 方法触发 resize 事件，或规定当发生 resize 事件时运行的函数。

9、each() 方法规定为每个匹配元素规定运行的函数。

提示：返回 false 可用于及早停止循环。

10、prev() 获得匹配元素集合中每个元素紧邻的前一个同胞元素，通过选择器进行筛选是可选的

11、angular中的ng-cloak防止页面闪烁

实例：//先用css定义ng-cloak

<style>

.ng-cloak{

display: none;

}

</style>

//在添加在html中就行

<body ng-app="module" ng-cloak class="ng-cloak">

或者哪里需要只要在那个标签添加属性就行

12、<!--HTML中form的属性novalidate，默认为false，阻止浏览器对表单进行验证-->

13、HTMl5的sessionStorage和localStorage – 统称web storage都是用于本地存储

* sessionStorage和localStorage的区别

sessionStorage ：用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此sessionStorage不是一种持久化的本地存储，仅仅是会话级别的存储。

localStorage ：用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。

* web storage和cookie的区别

1. web storage 是更大容量，Cookie的大小是受限的，并且每次你请求一个新的页面的时候Cookie都会被发送过去，这样无形中浪费了带宽，另外cookie还需要指定作用域，不可以跨域调用。
2. Web Storage拥有setItem,getItem,removeItem,clear等方法，不像cookie需要前端开发者自己封装setCookie，getCookie。
3. Cookie的作用是与服务器进行交互，作为HTTP规范的一部分而存在 ，而Web Storage仅仅是为了在本地“存储”数据而生。

* Html5中浏览器的支持

判断浏览器是否支持localStorage：

if(window.localStorage)

{alert("浏览支持localStorage")}

else{alert("浏览暂不支持localStorage") } //或者

if(typeof window.localStorage == 'undefined')

{ alert("浏览暂不支持localStorage") }

* localStorage和sessionStorage操作 – 具有相同的方法

1. setItem存储value

用途：将value存储到key字段

用法：.setItem( key, value)

代码示例：

sessionStorage.setItem("key", "value"); localStorage.setItem("site", "js8.in");

1. getItem获取value

用途：获取指定key本地存储的值

用法：.getItem(key)

代码示例：

var value = sessionStorage.getItem("key"); var site = localStorage.getItem("site");

1. removeItem删除key

用途：删除指定key本地存储的值

用法：.removeItem(key)

代码示例：

sessionStorage.removeItem("key"); localStorage.removeItem("site");

1. clear清除所有的key/value

用途：清除所有的key/value

用法：.clear()

代码示例：

sessionStorage.clear(); localStorage.clear();

* 其他操作方法：点操作和[]

web Storage不但可以用自身的setItem,getItem等方便存取，也可以像普通对象一样用点(.)操作符，及[]的方式进行数据存储。

具体代码：

var storage = window.localStorage;

storage.key1 = "hello";

storage["key2"] = "world"; //内部是通过json方式存储，[“”]方式访问里面的属性

console.log(storage.key1);

console.log(storage["key2"]);

* 遍历 -- localStorage和sessionStorage的key和length属性实现遍历

var storage = window.localStorage;

for (var i=0, len = storage.length; i < len; i++){

var key = storage.key(i);

var value = storage.getItem(key);

console.log(key + "=" + value); } //console.log()控制台输出普通信息

* 事件：当键值改变或者clear的时候，就可以触发storage事件

具体代码：

if(window.addEventListener){ window.addEventListener("storage",handle\_storage,false);

} //添加事件监听，调用handle\_storage函数

else if(window.attachEvent){

window.attachEvent("onstorage",handle\_storage);

}

function handle\_storage(e){

if(!e){e=window.event;}

}

监听的事件对象：

key String

oldValue Any

newValue Any

url/uri String

14、相对路径，绝对路径的写法

* 绝对路径

绝对路径是指文件在硬盘上真正存在的路径。那么如果要使用绝对路径指定网页的背景图片就应该使用 以下语句：

<body backround="E:\book\网页布局\代码\第2章\bg.jpg" >

* 相对路径

./ 当前目录 （不写的时候也表示当前目录）

../ 父目录

/ 根目录

所谓相对路径，就是相对于自己的目标文件位置。

在同一个目录:<body background="bg.jpg">

在其所在目录的“img”子目录里:<body background="img/bg.jpg"> (由于“img”目录是“第2章”目录下的子目录，因此在“img”前不用再加上“/”字符。)

在其所在目录的上级目录里:<body background="../bg.jpg"> (在相对路径里常使用“../”来表示上一级目录。如果有多个上一级目录，可以使用多个“../”.)

在其所在目录的上级目录里的“img”子目录里:<body background="../img/bg.jpg">