1.error() :抛出异常

2.parseHTML() :解析节点

3.parseJSON() : 解析JSON

4.parseXML() ：解析XML

5.noop() : 空函数

6.globalEval() : 全局解析JS

7.camelCase() : 转驼峰（内部）

8.nodeName() : 是否为指定节点名（内部）

9.each() : 遍历集合

10.makeArray() : 类数组转真数组

11.inArray() : 数组班indexOf

12.merge() : 合并数组

13.grep() : 过滤新数组

14.map() : 映射新数组

15.guid() : 唯一标识符(内部)

16.proxy() : 该this指向

17.access() : 多功能值操作（内部）

18.now() : 当前时间

19.swap() : css交换（内部）

Callbacks

1.add() 添加函数

2.fire() 执行添加函数

3.remove() 移出函数

Deferred 等同于 Callbacks

1.done() / progress()

2.resolve() / notify()

3.reject() 未完成状态

4.$.when() 多延迟对象

Support

浏览器功能检测

$(function(){

for(var attr in $.support){

$('body').append('<div>'+attr+':'+$.support[attr]+'</div>')

}

});

attr / setAttribute / prop / data

attr() 可以获取自定义属性

queue队列

1.queue() 入队

2.dequeue() 出对

3.队列中存的都是函数

Jquery选择器

1. find()

2.children()

3.next()

4.prev()

5.siblings() 上下同级所有 等同于 nextAll()和prevALL()

6.属性选择器

$('a[title^=num]').css('color','red');

7.内容过滤器

:contains(text) 选取含有”text”文本的元素，返回集合

:empty() 选取不包含子元素或空文本的元素，

8.可见性选择器

:hidden() 选取所有不可见

：visible() 选取所有可见

9.html()和text()方法

html() 获取/设置元素中HTML内容

text() 获取/设置元素中文本内容

val() 获取/设置表单中的文本内容

10.attr(key) 获取属性

attr(key,value) 设置key属性的值

11.offset() 获取某个元素相对于视口的偏移位置

12.position() 获取某个元素相对于父元素的偏移位置

13.scrollTop(value) 获取/设置垂直滚动条的值

14.scrollLeft(value) 获取/设置水平滚动条的值

15.内部插入节点的方法

append() 向指定元素内部后面插入节点content

append(function(index,html){}) 使用匿名函数向指定元素内部插入节点

appendTo(content) 将指定元素移入到指定元素content内部后面

prepend(content) 向指定元素content内部的前面插入节点

prepend(function(index,html(){})

prependTo(content)

16.外部插入节点的方法

after(content) 向指定元素的外部后面插入节点content

after(function(index,html){}) 使用匿名函数指定。。。后面插入节点

before(content) 向指定元素的外部前面插入节点

before(function(index,html){})

insertAfter(content) 将指定节点移到指定元素content外部的后面

insertBefore(content) 将指定节点移到指定元素content外部的前面

17.包裹节点

wrap(html) 向指定元素包裹一层html代码

wrap(element) 向指定元素包裹一层DOM对象节点

wrap(function(index){})

unwrap(html) 移除一层指定元素包裹的内容

wrapAll(html) 用html将所有元素包裹到一起

wrapAll(element) 用DOM对象将所有元素包裹在一起

wrapInner(html) 向指定元素的子内容包裹一层html

18.删除节点

remove() / detach()

都是删除节点，而删除本身后方法可以返回当前被删的节点对象，但区别在于前者在恢复时不保留事件行为，后者保留

empty() 删除掉节点里的内容

replaceWith() / replaceAll() 替换节点

19.表单选择器

:input 选取所有input,textarea,select,和button元素

:text 选择所有单行文本框

:password 选择所有密码框

:radio 选择所有单选框

:checkbox 选择所有复选框

20.bind() 绑定事件

unbing() 删除绑定事件

21.简写事件绑定方法

click() 单击

dblclick() 双击

mousedown() 鼠标按下

mouseup() 点击弹起后

mouseover() 鼠标移入

mouseout() 鼠标移出

mousemove() 鼠标移动

mouseenter() 鼠标穿过

mouseleave() 鼠标离开

keydown()

keyup()

keypress()

unload() 当卸载本页面时执行一个绑定函数

resize() 当改变大小时

scroll() 当滚动条拖动事件

focus() 获得焦点

blur() 失去焦点

focusin() 焦点激活

foucusout() 焦点丢失

select() 文本选定事件  
 change() 值改变

submit() 表单提交

22.复合事件

ready() 当DOM加载完毕触发事件

hover([fn1,]fn2) 当鼠标移入触发fn1，移出触发fn2

toggle(fn1,fn2[,fn3…]) 已废弃，当鼠标点击触发fn1，再点击触发fn2

23.阻止冒泡行为

e.stopPropagation()

24.阻止事件默认行为

e.preventDefault()

25. trigger() 触发被选元素的指定事件类型。

26.triggerHandler() 方法不会触发事件的默认行为，而trigger()会；

triggerHandler() 只会影响第一个匹配到的元素，而trigger()会影响全部

triggerHandler() 会返回当前事件执行的返回值，若无，返回undefined；而trigger返回当前事件触发元素的jquery对象

trigger()在创建事件的时候会冒泡。triggerHandler不会

27.live() 事件委托，具备动态绑定功能，原理是把事件绑定到祖先元素(document)上，只需要给$(document)绑定一次即可(已废弃)

28.delegate() live()的替代方法

$('#box').delegate('.button','click',function()

undelegate()

29.on() 代替 bind()方法 和 .delegate()/live()

on() 可以一次绑定多个事件

$('#box').delegate('click', '.button', function()

30.off() 解绑

31.one() 仅绑定一次

32.动画特效

show() 显示

hide() 隐藏

toggle() 切换

slideUp() 向上

slideDown() 向下

slideToggle()

fadeout() 淡出

fadeIn() 淡入

fadeToggle()

fadeTo() 到

33.自定义动画

animate()

34.Ajax (异步JavaScript 和XML)：无刷新状态跟新页面，实现异步提交

jQuery对Ajax采用了三层封装：最底层:$.ajax()

第二层: .load() , $.get() , $.post()

最高层:$.getScript() 和 $.getJSON()

35.load(url,(data,callback) )

36 Ajax

第一步： 创建XMLHttpRequest对象

第二部： 向服务器发出请求

open(GET/POST, url, asyns(true/false))

send(string:仅用于POST请求)

第三部：服务器相应

reponseText 获取字符串形式的响应数据

reponseXML 获取XML形式的响应数据

37. 全局事件方法是所有 Ajax 请求都会触发到，并且只能绑定在 document 上。而局部方法，则针对某个 Ajax

38. 工具函数

trim() 清楚左右空格

each() 遍历数组／对象

grep() 根据要求查找数组，返回数组

map()

merge() 合并数组

unique() 删除重复的数字／dom

isArray() 测试是否是数组

isFunction() 判断是否是一个函数

type() 判断数据类型

isNumeric(data) 判断数据是否为数值

isWindow(data) 判断数据是否为window对象

39. 默认验证规则 ----- Validate.js

required:true 必须输入字段

email:true 必须输入正确格式的电子邮件

<url:true> 必须输入正确格式的网址

date:true 必须输入正确格式的日期

number:true 必须输入合法数字（复数 小数）

digits:true 必须输入正整数

creditcard:true 必须输入合法的信用卡号

equalTo:’#pass’ 验证当前表单与id为pass表单内容是否相同

minlength ／ maxlength 输入最小／最大长度

rangelenngth[n,m]

range[n,m] 输入值介于n～m

min／max 最小／大值

remote:”check.php” 使用ajax方法调用check.php验证输入值

40.jQuery.UI.js

**1.dialog()的属性 对话框**

a．dialog(option) 以对象键值对的形式传参,每个键值对表示一个选项

b. dialog(‘action’,param) ，action是操作对话框的字符串，param是option的某个选项

内容属性：title 对话框标题； buttons 键值对方式添加按钮（键：名）

行为选项：autoOpen 是否自动打开；

draggable 是否可移动；

resizable 是否可改变大小；

modal：对话框外是否可操作

closeText 关闭按钮的文字

事件选项：focus 得到焦点时执行回调函数；

create 创建时执行

open 打开显示时执行；

close 关闭时执行；

beforeClose 关闭前执行 return false将不关闭；

drag 移动时执行；

dragStart 开始移动时

dragStop 移动结束时

resize 拉伸大小时

resizeStart 开始拉伸时

resizeStop 停止拉伸时

dialog（）方法

dialog(‘open’) 打开对话框

dialog(‘close’) 关闭对话框

dialog(‘destroy’) 删除对话框

dialog(‘widget’) 获取对话框jQuery整体对象

dialog(‘option’,param) 获取options属性的值

dialog(‘option’,param,value) 设置options属性的值

在dialog的事件中，提供了使用on（）方法处理的事件方法

**2.按钮UI**

button的属性：

disabled 按钮是否激活；

label 对用按钮上的文字

icons 对应按钮上的图标

text 是否显示文字

button(‘action’,param)方法

button(‘disable’) 禁用按钮

button(‘enable’) 启用按钮

button(‘destroy’) 删除按钮

button(‘refresh’) 更新按钮

button(‘widget’) 获取按钮jQuery对象

button(‘option’,param) 获取options属性的值

button(‘option’,param,value) 设置options属性值

**3.日历UI**

datepicker() 调用日历

datepicker()国际化选项：

dateFormat 指定日历返回的日期格式

dayNames 以数组形式指定星期中的天的长格式

dayNamesShort 。。。短格式

monthNames 以数组形式指定月份长格式

altField 为日期选择器指定一个<input>域

altFormat 添加到<input>域的可选日期格式

appendText 在日期选择器<input>域后面附加文本

showWeek 是否显示周

datepicker外观选项：

disabled 是否禁用

numberOfMonths 日历中同时显示月份个数，默认是1

showOtherMonths 是否填充空余单元格

changeMonth 是否显示月份的下拉列表

changeYear 是否显示年份的下拉列表

autoSize 是否自动调整控件大小

datepicker日期选择选项：

minDate 日历中可以选择的最小日期

maxDate 日历中可以选择的最大日期

defaultDate 预设默认选定日期，没有制定，则是当天

yearRange 设置下拉菜单年份的区间

**4.工具提示UI**

tooltip() 调用（先在title里写提示内容）

tooltip外观选项：

disabled 是否显示

content 设置title内容

items 设置选择器以限定范围

tooltipClass 引入class形式的css样式

tooltip位置选项：

position （键值对）my：目标左下角，at：以my为基准

show 是否淡入

hide 是否淡出

tooltip事件选项：

create 创建时调用

open 工具提示显示时调用

close 工具提示关闭时调用

**5.自动补全UI**

autocomplete({source:a,}) 调用方法（先创建数据源数组a）

autocomplete外观选项：

disabled 是否启用自动补全

source 指定数据源，本地／远程

minLength 触发补全列表最少输入字符数

delay 延迟显示

autofocus 设定为true，第一个项目会自动被选定

autocomplete事件选项

create

open

close

focus 当自动补全获得焦点时

select 当自动补全被选定时

change 当自动补全失去焦点且内容发生改变

search 当自动补全搜索完成后

response 当自动补全搜索完成后，在菜单显示之前调用

**6.邮箱自动补全**

邮箱格式： [用户名]@[域名]

autocomplete({

source:function(request,response){

request.term; 获取用户输入

response[数据源]

}

})

**7.选项卡UI—tabs**

tabs外观选项：

collapsible：是否允许选项卡折叠对应内容

disabled： 使用数组来指定禁用哪个选项卡

event： 触发tab的事件类型，默认为click

active： 初始化时默认显示哪个tab

heightStyle：默认content，根据内容伸展高度

show： 切换选项卡，是否淡入效果

hide： 切换选项卡，淡出效果

tabs()方法的事件：

create：当创建一个选项卡时触发

active： 当切换一个选项卡时触发

load： 当ajax加载一个文档后触发

**8.折叠菜单UI—accordion**

accordion外观选项：

collapsible：是否允许选项卡折叠对应内容

disabled： 使用数组来指定禁用哪个选项卡

event： 触发tab的事件类型，默认为click

active： 初始化时默认显示哪个tab

heightStyle：默认content，根据内容伸展高度

header： 设置折叠菜单的标题标签

icons： 设置想要的图标

Ajax表单插件---form.js

Form.js插件两个核心方法：ajaxForm() ajaxSubmit()

ajaxForm()提交方式

$(‘#reg’).ajaxForm(function(){

alert(‘’);

});

使用ajaxForm会直接实现ajax提交，提交的默认页面是form控件的action属性的值。

$(‘#reg’).submit(function(){

$(this).ajaxSubmit(function(){

alert(‘’);

)};

return false;

});

ajaxForm()是针对form直接提交的，所以阻止了默认行为，而ajaxSubmit()是针对submit()，所以要手动阻止默认行为。

resetForm：重置表单内容

clearForm：清空表单内容

beforeSubmit：提交之前执行，一般用于数据验证

success： 提交成功后执行

error：错误时执行

form.js工具方法

formSerializa()：表单序列化

fieldSerializa()：序列化某一字段

fieldValue()： 得到某个字段的value值

resetForm()： 重置表单

clearForn()： 清空表单

cookie插件

$.cookie(‘user’,’bnbns’) 生成一个cookie

$.cookie(‘user’,’bnbns’,{ 设置cookie参数

expires：7， 过期时间，7天后

path：‘／’ 设置路径，上一层

domain：‘www.dasha.com’设置域名

secure：true 默认false，是否使用安全协议https

})

$.cookie.raw = true; 默认false，是否关闭／打开解码