JQuery

session

http无状态 用户每次操作都要与服务器重新建立连接。下一个链接不能得到上一个链接的数据

PHP中为一个关联数组对象，存在服务器上，并保持任何数据，直到浏览器关闭才会没

session生命周期：

储存在cookie中,每个相应的页面都会有对应的编码，session结束为关闭浏览器才结束

session失效：1440s，24分钟后没任何操作session失效，内容整体失效

访问session

启动session: 调用session\_start()将session关联数组添加进内存中

session的使用：$\_SESSION[]=xx跟关联素组的使用方法一样

销毁Session

unset($\_SESSION[‘变量名’]);销毁一个变量

session\_destroy();销毁整个session

销毁session同时释放session在内存中的空间：session\_unset();session\_destory();

永久session: php.ini文件 session.gc\_maxlifetime = 9999

自定定时器ajax请求访问session

验证码：php随机生成4个数字，并放入session中，php将验证码绘制成图片返回，。

随机数rand(0,x，n) 从0-x中取n个随机数

.strlen() = .length()

substr($x, n,a) 取$x中n位置的字符a个

php图像生成：imagecreate(宽，高)设置图像颜色

imagecolorallocate(图片，r,b,g);设置图像颜色形态

imagefill(图片，0,0，颜色)设置使用那张颜色

imgreactangle(图片，0,0，宽的位置，高的位置，颜色)设置边框

imagesetpixel(图片，宽位置，高位置，颜色);

imagestring(图片，字体大小，x位置，y位置，值，颜色)

imagepng(图片)输出图片

imagedestroy(图片)销毁内存中的图片

读取text文件：file(文件名);返回按行显示内容的数组

什么是JQ

JQuery是第三方执行DOM操作的极简化的函数库

简化：DOM的增删改查、事件绑定、动画、AJAX

解决了大部分浏览器的兼容性问题

[www.jquery.com](http://www.jquery.com)

版本：1.x兼容旧版 2.x不兼容 3.x不兼容

新特性：for of 代替each

requestAnimationFarmer（）动画

Promise

废弃bind unbind delegate undelegated

所有代码运行在严格模式下

JQ的加载

CDM网络下有相应的src联网JQ引入

将JQ下载到网站本地文件夹，用相对路径加载

内容分发网络，客户端选择最近的服务器选择资源下载

原理：

JQ向全局添加了一个钟新类型JQ

创建JQ对象：

var jQuery = jQuery(‘selector’) = $(‘selector’)

.prototype = .fn

DOM对象都要用$()封装才能都使用jQery

JQAPI三大特性：

函数两用：括号内无值读取，有值更改

对JQ调用简化API等于对JQ里面的每个DOM对象调用相同的API

JQ下的选择器

css下的选择器JQ都能用

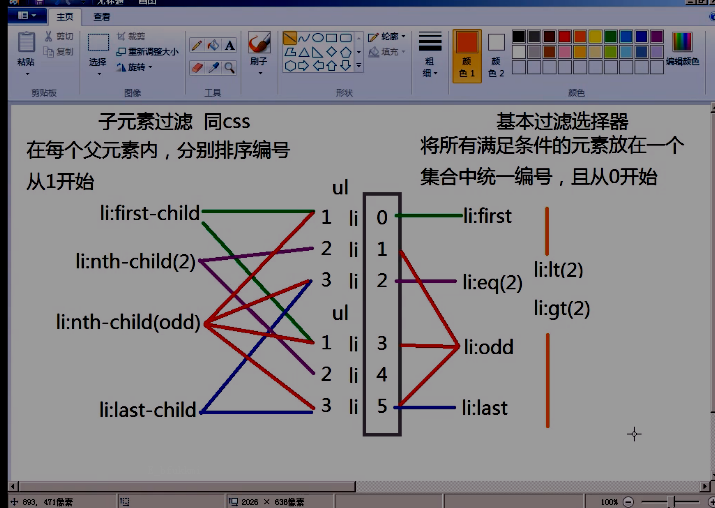
基本过滤选择器：：first/last

:eq(n)

:odd/even

:lt(n)小于

:gt(n)大于



可见性选择器：：hidden :visible

.innerHTML= xx == .html(xx);

.style.属性= xx == .css(属性，值/function(){return xxx});

var bool = $().is(“selector”)是不是这个选择器返回布尔

.show() display：block

.hide() display :hidden

查找：

内容过滤

:contains(text)以元素的文本内容为筛选条件

:has(选择器) 以元素的子元素特征作为筛选条件 class

:parent 非空元素

:empty 空元素

可见性过滤：

:hidden 选择所有隐藏元素 hidden display:none

:visible

表单元素过滤：

:input 选择所有表单元素

:type每种type的选择器 :text :password ：。。。

转态过滤

按节点间关系查找

$().parent()

$().children(‘selector’)

$().find(‘selector’)后代所有selector元素

$().children().last()/first()

$().next()

$().nextAll()之后所有

$().prev()

$().prevAll()之前所有

$().siblings()除我之外所有

修改：

内容：

.innerHTML = $().html()

.textContent = $().text()

.value = $().val()

$().empty 清空内容

属性：

HTML标准属性：$().attr(标准属性名，值) $().attr({xx:xx, xx:xx})

状态属性：$().prop(状态属性，boolean)

自定义扩展属性：$().data(‘属性名’，值) = dataset.属性名

样式：

修改多个样式，每个操作的API都会return $()所以可以通过链式操作

对于修改多个css可以$().css({属性：值， 属性，值})

.css()修改时会自动修改内联样式，获取时自动获取计算后的样式

$().addClass(‘’) 追加样式

$().removeClass(‘’)去除样式

$().hasClass(‘’)判断是否有样式

$().toggleClass(‘’) 有就去掉样式没有就加上样式

$().index()表示任何元素在其父元素下的位置

添加/删除、替换、克隆

1用HTML代码片段创建新元素

$a = $(‘HTML代码’);

2将新元素添加到DOM树

$parent.append($a)==$a.appendTo($parent)就返回值不一样

$parent.prepend($a) == $a.prepend($parent) 在元素前插入

$child.after($a) 将a插入到child之后

$child.before($a)将a插入到child之前

删除：$().remove()任何函数都可以remove自己

替换：$(旧).replaceWith($新)

$(新).replaceAll($旧)

克隆：$().clone() 仅克隆属性和样式，不克隆行为

$().clone(true) 都克隆

jQuery中$的原理：

$为一个工厂函数，内部自动new jQuery

1. 参数为选择器先查找DOM元素，在创建对象
   1. 优化同选择器与元素查找一样
2. 参数为DOM元素直接创建对象
3. 参数为HTML代码片段，创建DOM元素在创建对象

事件绑定

.addEventListener() = bind()

.removeEventListener() = unbind()

unbind(事件名，处理函数) 移除当前元素指定事件上指定的处理函数

unbind(事件名)移除当前元素上所有的处理函数

unbind()移除当前元素所有的事件

one()只触发一次，就移除

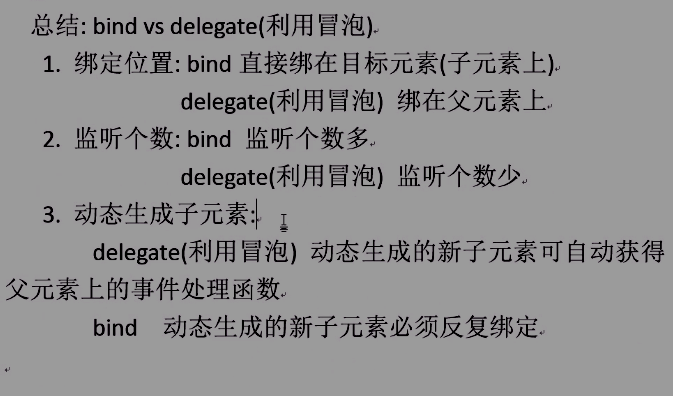
live/die()已废弃， 将所有元素的所有事件处理函数，集中绑定在document根节点上

事件委托：delegate()利用冒泡

$parent.delegate/undelegate(‘选择器’,事件名, function(){

this-》 e.target

})



on/off()

on(事件名、处理函数) 代替bind()

on(事件名，选择器(基于父级的选择器)，处理函数) 代替 delegate()

对常用事件名进行终极简化：

事件（处理函数）== on(事件，处理函数)

页面加载

DOMContentLoaded：当dom内容加载完后触发

$(document).ready(function(){})

简写：$().ready(function(){}) $(function(){})

等待完所有加载完成才触发：window.onload = function(){}

鼠标事件

mouseover mouseout 进出子元素触发父元素

mouseenter mouseleave 进出子元素不触发父元素

hover = 同时绑定鼠标进入和移出事件hover(function,function) 代替mouseenter 和mouseleave

模拟触发

$().事件名（）调用处理函数 == $().trigger(‘事件名’)没有函数就模拟触发

动画：

简单动画

显示隐藏：.show() .hide() .toggle() 不带参数默认使用display来眼熟隐藏，可带参数

效率：用定时器做的效果，效果写死 推荐用css

上滑下滑：slideUp() slideDown() slideToggle()

淡入淡出:fadeIn() fadeout() f adeToggle()

万能动画

$().animate({目标样式}， ms)

强调：支持单个数值的动画， 不支持颜色动画，用定时器实现

scrollTop 一个元素被隐藏的高度

$().offest()包含不变的top和left

排队和并发：

并发：多个css属性同时变化

放在一个animate内多个css，一起发生

排队：多个动画效果，先后顺序执行

后加多个animate（）就会按顺序实现

动画结束后自动执行：

$().animate({}, ms, callback(this))

停止动画：

$().shop() 判断是否在播放动画：：animated

shop()只清空当前动画， shop（true）清空所有动画

类数组对象操作：

$().each(function(I, elem){}) == foreach

$.each(类数组, fun)构造函数的each都可调用

$().index() == indexOf()

$(选择器).index(DOM/JQ)在左边找右边

$(选择器).index(选择器)在右边的结果和集中找左边

$(选择器).index()找其在其父元素下的位置

添加自定义函数：

向jquery.fn中添加新函数 = function(){this指将来调用子对象}

向jQuery原型对象中添加新函数，首先要验证是否已经引入了jQuery类型

官方插件：jQuery UI(了解，主要bookstracp)

插件/组件：具有独立样式和功能的小部件，方便重用，反复使用的功能和样式

下载，jQuery以来jQuery

效果：重写了animate支持颜色动画

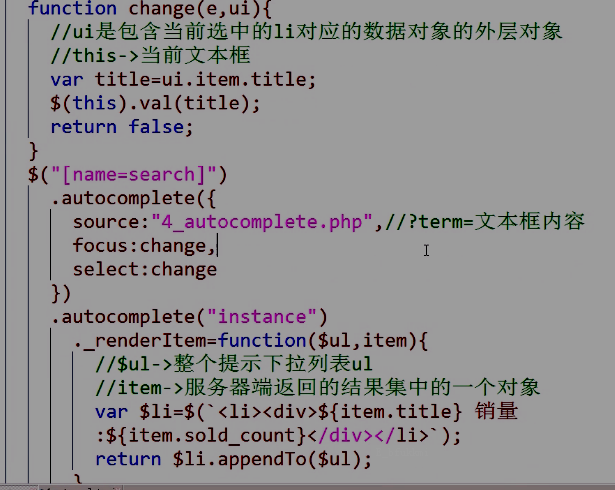
为简单动画API提供了更多的动画效果

重写了xxClass的动画效果 xxClss(类, 时间)

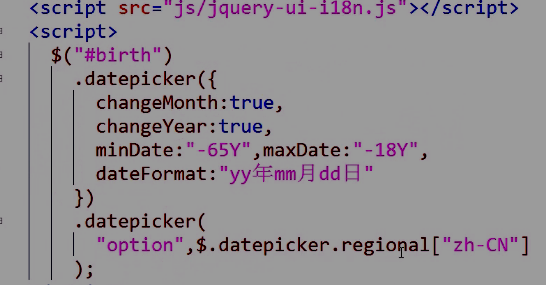
拖拽：$().draggable();(需要jquery-ui插件)

widgets: accordion()手风琴 需要有jQuery-ui.css

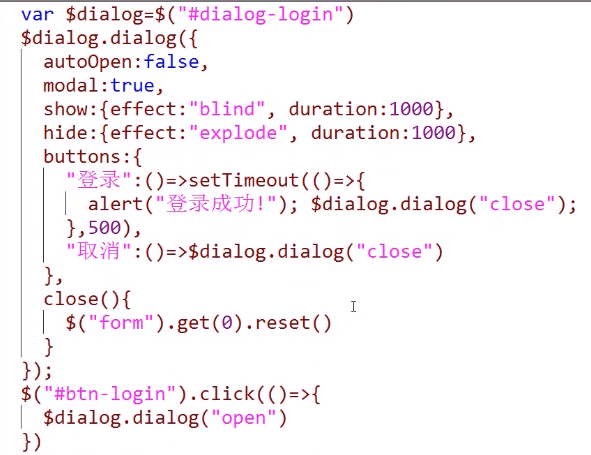
autocomplete 自动查询输入内容 .autocomplete({source:.php})会发送？term= ,并且接受战术名为label的json数组



$().datepicker({changeMonth:true, changeYear:true,minDate:,maxDate}) 日期组件



$().dialog（{autoOpen:false, modal:true, show:, hide: buttons:{xx:function…}, close(){}}）.(‘open’) 对话框



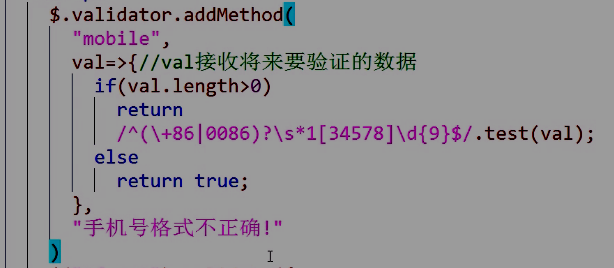
部件：定义HTML内容

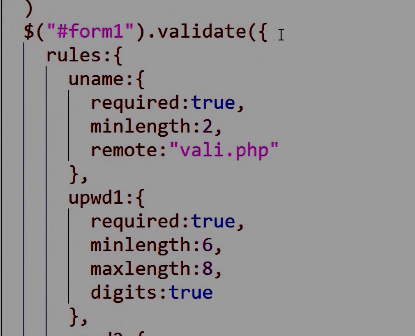
引入部件的css,js

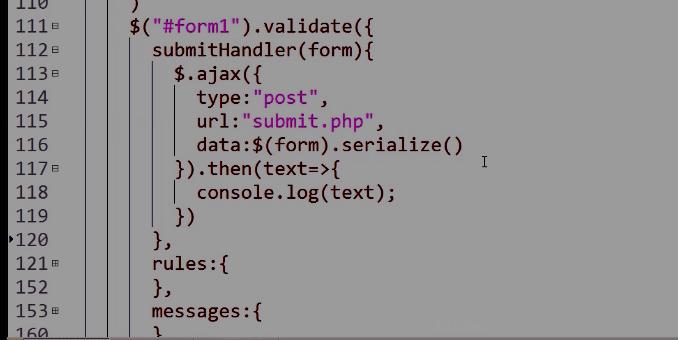
在父元素调用API和设置参数

第三方插件

validation







php接受数组必须在HTML的name设置为name[]

fileUpload

1. 普通文件上传
   1. HTML form method=post enctype=multipart/form-data
   2. php:apache隐式自动把文件放在temp目路下获得文件基本属性：（$\_FILES[xx]），将temp文件夹下的文件转储存:move\_upload\_file($xx[‘tmp\_name’], “文件夹”,$xx[‘name’])
2. ajax文件上传
   1. $.ajax({

url

type:post

data:new FormData(DOM)

contentType:false,转换为字符关

processData:false,数据处理关

xhrFields:{withCredentials:true}跨域请求不删除cookie

})

3、$().fileupload({

add(e,data){将文件添加到客户端的上传列表中的时候出触发此函数，data是贯穿于文件上传整个过程的一个数据对象

data.context=绑定每次添加文件的动作

}

done(e, data){上传完成时触发

data.context修改上传完成后的操作

}

})

wysiwyg：富文本编辑器

扩展：masonry瀑布流组件

封装自定义插件

jQuery侵入方式：

定义插件：

提出css

规定HTML结构

在jQuery的原型对象中添加插件函数

插件函数：侵入class和data-xx属性

为data-xx元素确定事件

引入js css

定义页面函数

$().插件函数（）

bootstrap方式：

引入插件css

定义页面结构，选择要用的css样式类

为元素添加自定义扩展属性，从而添加行为

引入插件的js

jquery中的ajax:

$.ajax({

type:get/psot,

<url:xxx>,

data:查询字符串/对象/formData,支持字符串又支持对象结构

datatype:’’ php设置了header可不写，但是php一定要写

success：function(data){}成功 调用此函数

error:function(data){}失败调用此函数 php出错

}).then(xx=>{})

简化：$.get(url, {data}).then()

$.post(url,{data}).then()配置文件只能用ajax写

$.getJSON()

$.getScript()

更简化。获得一段HTML片段，并填充到指定父元素下

$(父元素).load(‘url’,callback()) == $.get().then($().appendTo())

跨域请求：发送请求的起始域名和本地域名不一致

协议不同

一级域名，二级域名不同

端口号不同

ajax，xhr对象不允许跨域请求

解决：支持跨域：link script img iframe

方法一：script(可加载并执行js脚本)请服务端返回一段代码并执行 JSONP填充式json

缺点：重口难调；

方法二：在客户端提前定义处理函数，接受一个参数，将调用函数的语句放在服务器端返回

方法三：用？定义值=值传函数名传给服务器php然后将前后端分离

动态发起阔以请求：

动态创建script并追加到body结尾，并将script元素保存在公共的变量中

在处理函数结尾，自动删除保存的script元素

jQuery中的跨域

$.ajax({type:,url, data,datatype:’jsonp’}).then()

php文件中$\_REQUEST[‘callback’]

服务器端一句话可解决跨域：

Header（‘Access-Control-Allow-Origin:\*’）；

$().offset.top/$().offset.leftDOM相对于整个制作的网页上与左的位置

iconfont 字体的图标 可以用字体的css样式

css和svg文件

clss=’iconfont xxxxxx’

$(window).ready(() => {}) === windw.onload()不过wondow.onload只能触发一次，而jquery的可以用多次