正课:

1. \*\*\*自定义插件:

2. Ajax

3. \*\*\*\*\*跨域访问:

1. \*\*\*自定义插件:

前提: 已经用html,css,js实现了,发现以后有用，就封装成插件

2种风格:

1. jQueryUI侵入式: 5 04 10 5 04 11

1. css应提取到独立的css文件中

强调: 所有插件内的子元素选择器都要以插件父元素class开头，避免与其他插件冲突。

2. 定义独立的js文件：

1. 判断是否引入jQuery

2. 在jQuery的原型对象中添加插件函数

1. 侵入class和自定义扩展属性

2. 绑定事件

3. 使用插件:

引入插件.css,

按插件要求编写HTML内容

引入jQuery

引入插件.js

编写自定义脚本: 查找插件父元素，调用插件API

2. Bootstrap DIV式: v3.bootcss.com 5 04 12 5 04 10

1. css应提取到独立的css文件中: 同jQueryUI方式

2. 定义独立的js文件：

强调: 不需要添加函数，不用侵入

3. 使用插件:

引入插件.css

按插件要求编写HTML内容

在HTML元素上，手动添加class和自定义扩展属性

引入jQuery

引入插件.js

强调: 不需要调用任何插件函数，jq调用函数是因为不用手动添加属性

2. Ajax: *tuddsec 5 1 关键词查询*

$.ajax({

type:"get/post",

url:"data/1\_menu.php",

data: 查询字符串/{ 参数:值, 参数:值, ...}/$form.serialize(),

dataType:"json"//如果返回json，则自动转为对象,

success:function(data){ //请求响应结束且成功

},

error:function(){ //请求响应结束，但出错

},

complete:function(){ //只要请求响应结束

}

})//3.x .then(function(data){ ... })

dataType: 如果服务端定义了响应的内容类型为applicaton/json， 则$.ajax可自动转换，不必写dataType

几种简写：

$.get("url", data, function(data){...}, dataType)

$(parent).load("url",function(){ ... })请求一片段,填充parent内 5p63:30s 16 project02 header.js

$.getJSON() *省dataType*

$.getScript() *如果服务器发回来****一句话/脚本***

$.post(一样)

3. \*\*\*\*\*跨域请求: 5p 120:00

什么是跨域:

1. 域名不同， 2. 二级域名不同, 3. 端口号不同, 4. 协议

问题: 服务端不允许用xhr对象跨域发送ajax请求

解决: 利用可以跨域的元素(link, script, iframe, img, ... )（cdn网络拿jquery，网络图片）

1.用script变通实现跨域请求: 5p weather

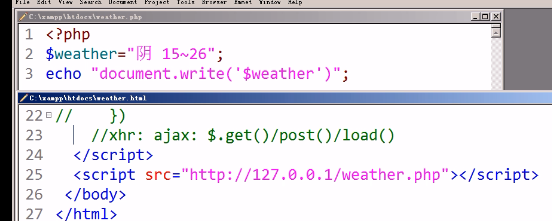
JSONP: JSON with padding 填充式json

方案1:

服务端: 将返回的数据，填充在一条js语句中返回

客户端: 用script src="http://xxx/xx.php"

问题: 客户端数据的处理逻辑是在**服务端写死的**，众口难调

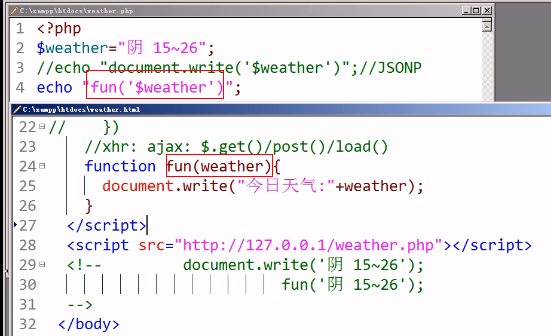


方案2:

服务端: 返回一条函数调用的js语句，将数据作为参数

客户端: 提前定义和服务端同名的js函数

问题: 函数名是在服务器端定死的



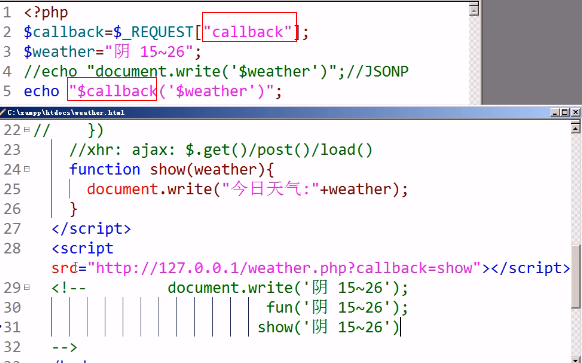
方案3: 5p weather (访问hostdcuments的weather.php)

服务端: 先接受客户端传入的函授名参数(callback)

在返回结果中动态拼接函数名

客户端: script src="xxx.php?callback=函数名"

问题: script是写死的，只能在页面加载时发送一次请求



方案4: 5p weather (访问hostdcuments的weather.php)

客户端: 不要写死script

在处理函数中，动态创建script，发送请求

在处理函数结尾，删除动态创建的script



问题: 和传统ajax不一致！

方案5: jsonp echo "$callback('$weather')"

$.ajax({

type:"",

url:"",

data:...,

dataType:"jsonp",

success:function(){

}

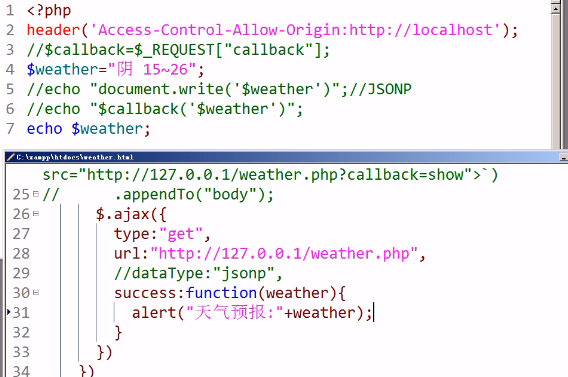
})



服务器端返回的是随机函数名包着的数据参数

2. 通过服务器端开放跨域: 5p weather.php

php: header('Access-Control-Allow-Origin:http://请求来自的网址');



--------------------------------------------------16 project 笔记

Session 5p214 16 project ——data——user——三个php

**欢迎回来例子**：5p 245 header.js 结构uname 放进.html()

什么是: 服务器端存储用户信息的存储空间

为什么: 普通的http通信是没有状态的

前后两个通信之间无法保存/共享数据

何时: 希望前后两个http通信之间能共享用户数据时

生命周期:

1. 打开浏览器首次访问网站时，立刻创建

服务端返回唯一的sessionid，对应服务端一个专属的存储空间

2. 只要浏览器窗口不关，今后每次访问服务器端都会自动携带sid到服务器端

3. 服务器端: php

向自己的session中存入一个变量:

session\_start();

$\_SESSION["变量名"]=值

从自己的session中取出变量:

session\_start();

$变量=$\_SESSION["变量名"]

注销:

session\_start();

session\_unset();

session\_destroy();

4. 关闭浏览器后，sessionid自动失效！

下次再访问网页，重新分配sessionid和存储空间。