正课:

1. 插件:

2. ajax:

3. 跨域:

1. 插件: 也称组件

什么是: 拥有专属的HTML，CSS和js的独立页面区域

为什么: 重用！

何时: 只要一个功能/区域可能被反复使用时

如何:

3个来源:

1. 官方插件：jquery ui

依赖于jQuery, 必须先引入jQuery，再引入jQuery UI

如何:

1. 引入jQuery UI的css

2. 按组件的要求，自行编写HTML结构和内容

3. 引入jQuery.js, 再引入jQuery UI.js

4. 查找插件的父元素，调用jQuery UI的插件函数

$(插件的父元素).插件函数()

侵入性: 插件根据自己的需要，自动添加样式类或自定义扩展属性

2. 第三方插件：

3. 自定义插件：

定义:

前提: 已经用HTML，CSS，JS实现了插件的效果和功能

1. 将插件的CSS提取出来，保存在一个独立的css文件

要求: 为了避免和其他插件存在相同的样式类发生冲突，必须保证每个样式类都要以统一的插件类名作为前缀！

2. 定义插件的js: 向jQuery的原型对象中添加自定义插件函数

jQuery.fn.插件函数=function(){

//this->$(父元素)

//2件事:

//1. 侵入class:

//2. 绑定事件处理函数:

}//调用时: $(父元素).插件函数();

使用插件: 同使用jQueryUI的做法

1. 引入插件的css

2. 按照插件要求编写html

3. 引入jquery.js和插件.js

4. 查找插件父元素，调用插件函数

2. ajax:

$.ajax({

url:"服务端接口地址",

type:"get/post",

data:{ 参数1:值1, ... },

dataType:"json", //可自动将服务器返回的json字符串转为对象

success:function(res){//onreadystatechange 返回响应，且响应成功时自动触发

//res会自动获得服务端返回的数据

//用res执行DOM操作

}

})//jquery 3.x 支持Promise

.then(function(res){

})

3. 跨域:

什么是: 一个域名下网页，向另一个域名下发送请求，请求另一个域名下的资源

比如: 现在在http://localhost/index.html下

<script src="http://www.jquery.com/jquery.js"

<img src="http://tmooc.cn/stylesheet/img/logo.png"

<link rel="stylesheet"

href="http://v4.bootcss.com/bootstrap.css"

问题: ajax的xhr对象，禁止发送跨域请求

包括:

1. 一级域名不同: www.a.com -> www.b.com

2. 二级域名不同: oa.tedu.com -> hr.tedu.com

3. 端口不同: localhost:5500 -> localhost:3000

||

4. 协议不同: http:localhost -> https:localhost

80 443

5. 即使同一台机器: 域名 -> IP

localhost 127.0.0.1

如何发送异步跨域请求:

1. JSONP: JSON with Padding 填充式json

问题1: ajax不能发送跨域请求

解决1: 请<script>元素帮助发送请求

问题2: <script>发送请求，必须返回一条可执行的js语句

解决2: 修改服务端res.write()，其中，将要返回的数据，填充进一条可执行的js语句中，一起返回。

问题3: 服务端返回的js语句是写死的，众口难调

解决3: 在客户端定义一个处理函数

function show(res){

//对res执行任何想要的操作

}

服务端返回一条函数调用语句，函数名必须和客户端定义的函数名保持一致:

res.write(`show('${weather}')`)

返回: show('晴 -10~-2 from dong')

在客户端执行时:

调用show函数:

参数res自动得到了'晴 -10~-2 from dong'

实现了: 客户端操作与服务端的分离

问题4: 服务端将函数名规定死，也是众口难调

解决4: 客户端发送:

<script src="url?callback=客户端函数名"

服务端:

先获得req中callback变量中的函数名

再将函数名拼接到res.write()中，代替写死的函数名

实现了: 处理逻辑和函数名与服务端无关

问题5: <script>在客户端只写死一次，仅能在首次加载页面时执行一次，无法反复发送请求

解决5: 动态添加<script>元素

在单击事件中:

$('<script src="http://localhost:3000?callback=doit">').appendTo("body");

问题6: 新增的<script>无法自动删除，造成积压

解决6: 在自定义的回调函数结尾，查找最后一个script，删除。

其实: $.ajax可自动实现jsonp效果:

$.ajax({

url:"url",

... : ... ,

dataType:"jsonp", //使用jsonp方式请求服务端

success:function(res){ ... }

})

原理: 同以上6步:

1. 动态创建script元素发送请求

2. 自动为success匿名函数定义随机函数名拼接到url?callback=随机函数

3. success函数执行后，自动删除script元素

强调:jsonp单靠客户端无法实现，必须服务端负责拼接函数名和要返回的数据。所以必须客户端服务端同时修改才可支持。