正课:

1. 什么是jQuery

2. 如何使用jQuery

3. \*\*\*JQuery对象

4. 查找

5. 修改

1. 什么是jQuery:

第三方的 极简化的 DOM操作 函数库

为什么: 2个主要原因:

1 . 是DOM操作的终极简化:

1. 增删改查

2. 事件绑定

3. 动画

4. Ajax请求

2. 解决了绝大部分浏览器兼容性问题:

凡是jQuery让用的，都没有兼容性问题!

何时: 今后几乎所有的项目，框架。。。都是用jQuery开发的。

2. 如何使用jQuery:

快速学会一项技术3步:

1. 上官网，看是什么

2. 看快速入门

3. 看手册

官网: jquery.com

快速入门: 官网首页

手册:

下载:

版本: 1.x 支持旧浏览器

未压缩版:

优: 可读性好，具有完备的注释，格式以及见名知义的变量名

缺: 体积大

何时: 开发或学习时

压缩版:

特点: 1. 去掉所有空字符

2. 去掉所有注释

3. 极简化变量名

优: 体积小，便于传输

缺: 毫无可读性

何时: 生产环境

2.x 不再支持旧浏览器

3.x 不再支持旧浏览器,增加了新特性!

鄙视: jq3.x支持哪些新特性:

1. 所有API默认运行在严格模式下

2. 支持for...of，代替each

3. 新动画API : requestAnimationFrame()

4. 支持promise

引入: 2种:

1. 将js文件保存在项目本地文件夹中，在页面用script src="相对路径"

2. 引用CDN网络上的js文件

什么是CDN: 内容分发网络: 可根据客户端到服务端的网络状况的优劣，自动选择最优的服务端下载资源。

为什么: 如果文件仅保存在一个位置的服务器上，距离较远的客户端，访问就慢

何时: 如果项目中的共享文件，需要各地用户都能快速访问，就必须放在CDN网络上。

如何: 不用下载所需资源文件

去官网找CDN网址:

在页面中用官网提供的script网址，引入CDN中的资源文件

3. jQuery对象:

引入jquery.js: 向window中添加一种新的类型jQuery

包括: 1. 构造函数/工厂函数: 用于创建该类型的子对象

2. 原型对象: 保存该类型的子对象才能共用的API

什么是jQuery对象: 封装找到的所有DOM对象的类数组对象。并提供了操作DOM对象的简化版API

为什么: jQuery简化版API，只能被jQuery类型的子对象使用。普通DOM对象不能使用!

何时: 今后，只要想使用jQuery简化版API，都必须先创建jQuery类型的子对象

如何:

创建:

1. 用选择器查找，并创建jQuery对象:

var $obj=jQuery("选择器")

2件事: 1. 创建jQuery类型的子对象

2. 用选择器查找要操作的DOM对象，装入jQuery对象中

将来: 对jQuery对象调用任何简化版API，等效于自动对其中的每个DOM元素都调用一遍API——自带遍历

何时: 在暂时没有获得任何DOM对象时，都用选择器获得第一个要操作的DOM对象

2. 将已经获得的DOM元素，封装为jQuery对象

var $obj=jQuery(DOM对象)

何时: 如果通过this,e.target已经获得一个现成的DOM对象

其实所有jQuery都可替换为$

jQuery API 三大特性:

1. 自带遍历效果:

$btns.click() 等效于对$btns中每个btn自动调用click

2. 一个函数两用:

$elem.html( ['新内容'] )

没给新内容，默认是读取原内容

给了新内容，就变为修改!

$elem.css("属性名"[,新属性值])

3. 多数API都返回正在操作的jQuery对象:

为什么: 复用jQuery对象，避免反复查找和反复创建对象

学jQuery还是在学DOM:

构建DOM树->查找触发事件的元素->绑定事件->查找要修改的元素->修改元素(内容,样式,属性)

4. 查找: 2种:

1. 按选择器查找:

何时: 在没有获得任何元素情况下，找第一个

选择器: jQuery支持所有CSS选择器

2. 按节点间关系查找:

何时: 已经获得一个元素，找周围的元素

包含:

1. 父子:

$elem.parent() => elem.parentElement

$elem.children(["selector"]) => elem.children

$elem.children().first() => elem.firstElementChild

$elem.children().last() => elem.lastElementChild

2. 兄弟:

$elem.prev() => elem.previousElementSibling

$elem.next() => elem.nextElementSibling

5. 修改:

内容: $().html(['新内容的HTML片段'])

属性: 3种:

标准属性:

$elem.attr("属性名"[,"属性值"]) => elem.属性名

状态属性: elem.属性名

$elem.prop("属性名"[,bool]) => elem.属性名

自定义扩展属性: elem.dataset.属性名

样式:

$().css("属性名"[,新值])

问题: 一次只能修改一个css属性

解决: .css({ css属性名:值, ... })

问题: 如果一次修改的样式过多，还需要撤回，代码重复会很多

解决: 用class，批量应用/撤销样式

$().addClass("class1 class2 ...")

$().removeClass("class1 class2 ...")

$().hasClass("class1")