正课:

1. 什么是jQuery 3特点 第三方 简化方面 使用原因

2.使用jQuery 3了解 版本 引入 原理

3. 查找方法` 2掌握 选择器 节点关系

4. API 3个规律

1. 什么是jQuery

第三方开发的 执行DOM操作的 极简化的 函数库

**第三方:** 下载

执行DOM操作: 学习jQuery，还是在学DOM

**极简化**: jQuery是DOM操作的终极简化: **4个方面**:

1. 增删改查

2. 事件绑定

3. 动画

4. Ajax

函数库: jQuery中一切都是函数

**使用jQuery2个原因:**

1. 终极简化

2. 解决了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

如果不兼容，jQuery在底层用程序模拟实现

2.使用:

3步:

1. 上官网看是什么:

2. 看快速入门:

3. 下载手册，备查:

下载: 版本:

1.x 兼容旧浏览器

未压缩: jquery-1.11.3.js

具有: 完备的注释，代码格式和见名知意的变量名

可读性好，体积大——开发和学习阶段使用

压缩: jquery-1.11.3.min.js

移除所有注释, 清除所有空字符格式，极简化变量名

体积小，可读性差——生产环境

2.x 不再兼容旧浏览器

3.x 也不兼容旧浏览器，还增加了新功能

1. 所有代码运行在严格模式下

2. 支持for...of，代替原有each()

3. 新动画API: requestAnimationFrame()

4. 支持Promise

引入: 2种:

1. 将js文件下载的项目本地

2. 引入CDN网络中的js文件

CDN网络: 内容分发网络

可根据客户端据不同服务器的网络状况，自动选择最优的服务器下载资源

关键词：jQquery对象 构造函数 原型对象Api

原理:

*全局创建jquery自己的类型，将选择器查找的dom元素类数组对象封装成jquery对象（也就是封装在新建的jQuery构造函数中）新建构造函数最终简化成用$（也就是工厂函数），然后为这些jQuery对象提供保存在原型对象中的共同方法api。从而实现了dom查找和操作dom元素的功能）*

引入jquery.js，其实是在**全局**创建了一种新类型: 2部分

1. 构造函数: 创建jQuery类型的子对象

2. 原型对象: 保存所有jQuery类型的子对象共有API

创建jQuery类型的子对象:

什么是jQuery类型的子对象: 专门封装一个或多个DOM元素，并提供操作DOM元素的简化版API

本质是一个类数组对象。

为什么: 普通DOM元素不是jQuery类型，不能用简化版API

只有将找到的DOM元素，放入jQuery对象中，才能使用简化版API

何时: 只要想使用jQuery的简化版API，都要先创建jQuery对象

如何: 2种:

1. 用选择器查找DOM元素，并创建jQuery对象:

var $elems=new jQuery("selector")

先找到符合selector要求的DOM元素,再创建一个jQuery类型的子对象，保存找到的DOM元素

简化: var jQuery=function("selector"){//工厂函数变成jQuery

return new 构造函数("selector")

}// jQuery("selector") 等效于 new 构造函数()

更简化: window.jQuery = window.$ = jQuery;//jQuery变成$

//调用$()等效于调用jQuery()

*创建dom对象：window.jQuery = window.$ = jQuery=new jQuery()=>构造函数jQuery*

对原型对象的简化: jQuery.fn = jQuery.prototype = {

jQuery.fn也是jQuery类型的原型对象的别名

2. 直接将获得的DOM元素封装进jQuery对象:

var $elem=$(dom元素对象);

3. 查找:

1. 按选择器查找:

jQuery支持所有CSS选择器，并扩展了部分自有的选择器

基本选择器: 5个 同CSS

#id element .class \* selector1,selector2,...

层级选择器: 4个 同CSS

parent child parent>sub prev+next prev~nextsiblings

子元素过滤: 4个 同CSS

特点: 在各自父元素内分别编号，从1开始

:first-child :last-child :nth-child(n) :only-child

基本过滤(位置过滤): 7个 jQuery中新增

特点: 先取出**所有**符合条件的**元素放入一个集合**中统一编号。编号从0开始

:first/last :even/odd偶奇 :eq/gt/lt(i) 等大小

属性过滤: 同CSS 1 8？

[属性名]

[属性名=值]

[属性名^=值] 属性的值以...开头

[属性名$=值] 以...结尾

[属性名\*=值]

[属性名!=值]

input.txt[title=用户名]

何时:代替class选取多个元素，添加行为 data-

可见性过滤: 2个: 1 9

:visible :hidden(display:none input type="hidden")

内容过滤: 2组:

1. 用子元素或内容的特征判断父元素: 1 10

1. 以**内容**文本作为特征: :contains(text) $("button:contains(提交) ").css()

2. 以**子元素特征**: :has(选择器)

$("div.alert:has(.close) ").css()/$("div.alert:not(:has(.close) )").css()

2. 以**是否有**子元素或内容为条件:

:parent 非空

:emty 空

表单元素过滤: 1 13

:input 选择所有表单元素(input select textarea button)

每种type都有专门的选择器:

:text :password :radio **:checkbox** :file :submit...

状态过滤: :enabled :disabled :checked :selected

2. 按节点间关系查找:

2大类:

1. 父子:

$elem.parent() 父元素

$parent.children() 直接子元素

$parent.children().first()

$parent.children().last()

2. 兄弟:

$elem.next() 后一个兄弟

$elem.prev() 前一个兄弟

$elem.is("selector") **判断**$elem是否符合selector选择器的要求 1 9

*if($ul.is(":hidden")){$ul.show()};*

$elem.show()/hide() **代替**d**isplay**="block"/"none"

$elem.css("CSS属性名","属性值") style.css属性名=值

$elem.html("内容") innerHTML="内容"

jQuery API三大规律:

1. **自带遍历**: 对整个jQuery对象调用API等效于对其中每个DOM元素都调用一次**API**

2. 一个**函数两用**: 给新值，就修改。没给新值，就返回旧值 （.html()）

3. **链式操作**：前一个函数要**返回当前正在操作的jQuery对象本身**（dom元素）让下一个函数用。 （jQuery 链式操作原理 1 9 16:49）