第一天

课程体系：

一、javascript js

时间： 12天

内容：

javascript 8天

jquery 3天

bootstarp 1天

作用： 就是给网站加特效

二、PHPCMS 二次开发

时间 3天

作用：

快速建站

三、非关系型数据库

时间 3天

内容：

memcache 1天

redis 2天

作用：

给网站加缓存

四、node.js 长连接

时间：4天

内容：

node.js 1天

express 3天

作用：

实现项目中的长连接（聊天）

五、laravel 框架

时间：6天

六、项目

时间 20天

七、 特级课

就业期

mysql优化

微信开发

thinkphp3.2

yii2.0框架

5就业冲刺

面试题

扩展

=====================================

javascript：

发展史：

美国网景公司 发布 javascript 其实 “livescript”

使用ECMA语法标准

'ECMA5' --> ECMA6

ECMA 组织

其中有一个原因： 兼容性

合作： sun

java ---》 javascript

因为： ajax 无刷新技术

=====================================

任务清单：

1.html javascript -- 引入

2.html javascript -- 变量的声明

3.html javascript -- 数据类型

4.html javascript -- 数据类型的注意事项

5.html javascript -- 数据类型的转换

6.html javascript -- 运算符

7.html javascript -- 计算相加100结果

======================================

一、javascript 是什么？

JavaScript是一种基于对象和事件驱动并具有安全性能（由浏览器直接解析执行）的脚本语言。

二、javascript 特点：

可以和html混编

兼容性

安全性

跨平台

目标：

特效

表单验证

ajax

三、使用场景：

3.1 引入：

外部：

<!--

type 类型

src 引入文件的url

注意：

带有src的script 标签内部不允许出现js任何代码

-->

<script type="text/javascript" src="./1.js"></script>

内部：

<script type="text/javascript">

js代码

</script>

内嵌：

<!-- on 动词 click 点击 -->

<button onclick="alert('删除成功')">点击删除</button>

3.2 变量声明

使用关键字 var

格式

var 变量名 = '值'; 局部变量

变量名 = '值'; 全局变量

注意：　推荐使用var声明

变量作用域：

局部：

在函数体内部有效，局部变量的声明周期就是从函数的调用开始到函数的结束

全局：

全局变量的生命周期是从这个脚本的开始到这个脚本的关闭

变量声明的注意事项

1. 可以使用数字、字母、下划线但是：数字不能开头

2. 变量名严格区分大小写

3. 不能使用关键字

4. 变量名有意义

调试方式：

alert() 弹框

document.write() 在html页面中输出

console.log() F12 控制台输出

3.3 注释和分隔符

注释：解释说明

单行 //

多行 /\*\*/

分隔符：

; 推荐

3.4 js数据类型

js数据类型（6种）

字符串

'' ""

数值型

整型

浮点

NaN not a number

布尔

true false

对象

数组

对象

null

未定义

undefined

方法

function

3.4 数据类型注意的事项

1. 单引号和双引号都不解析变量

2. 在js中永远不要对一个浮点数进行精确的计算 使用toFixed()保留小数位

3. NaN (not a number) 和任何数据进行计算，最后都是NaN,检测一个值是否是NaN 使用isNaN()

3.5 数据类型的转换

parseInt() 将指定的数据类型转换成number型 整型

parseFloat() 将指定数据类型转换成number型 浮点

主意：parseFloat() 规律和parseInt()一致

规律：

1. 如果是纯数字的字符串 ，则返回整体

2. 如果是数字开头，则返回第一个非数字前面部分

3. 如果是非数字开头，则返回NaN

3.6 运算符

+ +=

规则：

+ +=

如果运算符俩端是数值型，则用于计算

如果运算符有一端是字符串，则用于连接

3.7 流程控制

if(true){

}else if(){

}else{

}

for(var i = 0;i<10;i++){

console.log(i);

}

switch(str){

case 1:

break;

case 2:

break;

default:

}

获取html元素

document.getElementById()

===================================

作业：

基础作业：

代码敲3遍

扩展：

1. 使用js实现99乘法表

2. 使用js实现计算器 优化：使用一个函数完成

第二天

任务清单：

1.html javascript --- 99乘法表

2.html javascript --- 计算器

3.html javascript --- 函数的声明

4.html javascript --- 函数的特点

5.html javascript --- 系统函数

6.html javascript --- html样式、文本、值

7.html javascript --- 进度条

===============================================

课堂笔记

一、函数的声明

1. 普通方式

function fun(){}

2. 匿名函数

function (){}

触发：

1. var s = function (){}

s()

2. (function (){})

()

3. btn.onclick = function(){}

3. 关键字声明

new Function();

二、函数的特点

1. 变量作用域

全局变量： 全局变量的声明周期就是从这个脚本的开始到脚本关闭

局部变量： 局部变量的声明周期就是从这个函数的开始到这个函数的结束

2. arguments 系统关键字 以数组的形式 接收所有的实参

3. 默认值 在函数中 形参加上默认值 会有兼容性

4. 返回值 在函数中没有返回值 默认返回undefined

三、系统函数

clearInterval()清除定时器

setInterval() 定时器

eval() 将一段字符串当成js脚本来执行

decodeURI() 将url的编码反解为中文

encodeURI() 函数可以把字符串作为URI进行编码

location.href 获取当前url

四、html文本 、默认值 、css样式

style 设置元素的css样式

格式

obj.style.css属性= '属性值'

注意：

如果css属性是复数形式，把中间的‘-’去掉，然后后面的属性使用首字母大写驼峰命名法

value 默认值

格式：

object.value 获取

object.value = '' 设置

针对于form表单中元素

innerHTML 文本值

格式：

object.innerHTML 获取

object.innerHTML = '' 设置

第三天

任务清单：

1.html javascript --- 对象的声明

2.html javascript --- 对象的遍历

3.html javascript -- 字符串对象

4.html javascript -- 动态替换字符串

5.html javascript -- 数组

6.html javascript -- 数组的注意事项

7.html javascript -- 时间对象

======================================================

课堂笔记：

一、对象的声明

1. 使用关键字声明

var obj = new Object;

obj.name = '';

obj['sex'] = '';

2. 使用json格式

var obj = {'key':'val'};

3. 使用构造函数

function obj(){

this.name = '';

}

var p = new obj();

对象访问

.

[]

this

二、对象遍历

for() 遍历数组

for..in 遍历对象

with(obj){

}

三、内置对象

字符串对象

声明

var str = new String();

var str = '';

方法

trim() 清除两侧的空白

charAt() 获取字符串指定位置上的字符

indexOf() 获取指定字符或者字符串首次出现的位置

lastIndexOf() 获取指定字符或者字符串最后一次出现的位置

concat() 连接两个或者多个字符串

slice() 截取字符串 从开始到结束 要前不要后

split() 按照指定的字符或者字符串来拆分字符串数组

substr() 截取字符串

toLowerCase() 将字符串转化成小写

toUpperCase() 将字符串转化成大写

replace() 替换一个与正则表达式匹配的子字符串

数组对象

声明

var arr = [];

var arr = new Array()

函数

concat() 连接一个或多个数组

join() 将数组元素连接起来转化成一个字符串

reverse() 翻转数组的元素

slice() 返回数组的一部分

sort() 对数组元素进行排序

splice() 插入、删除或替换数组的元素

push() 从数组的尾部压入元素

pop() 从数组的尾部删除元素

unshift() 从数组的头部插入一个元素

shift() 从数组的头部删除一个元素

数组的注意事项

1. 数组不能使用[]直接赋值

2. 下标有连续性

3. 没有关联数组

4. 二维数组使用

var a = [];

a[0] = [];

a[1] = []

[

[],

[],

[],

]

时间对象

实例化

var date = new Date;

方法

getFullYear() 获取年份

getMonth() 获取月份

获取当前的月份 +1

getDate() 获取天

getHours() 获取小时

getMinutes() 获取分钟

getSeconds() 获取秒

getDay() 获取星期

0-6 对应着星期日至星期六

数学对象

Math

round() 浮点数四舍五入

max() 获取最大值

min() 获取最小值

abs() 绝对值

floor() 舍去取整

ceil() 向上取整(进一取整)

pow() 获取n次幂

sqrt() 求平方根

random() 获取0-1之间的随机数

第四天

任务清单

1.html javascript-- 正则基本使用

2.html javascript -- 表单验证

=====================================

课堂笔记

一、正则对象

1.1 声明

var preg = /abc/模式修正符; //声明正则不加''

var preg = new RegExp("abc",模式修正符);

1.2 基本语法

元字符

. 查找单个字符，除了换行和行结束符

\* 匹配任何包含零个或多个 n 的字符串。通配符 匹配所有

+ 匹配任何包含至少一个 n 的字符串

? 匹配任何包含零个或一个 n 的字符串。

n{X} 匹配包含 X 个 n 的序列的字符串。

n{X,Y} 匹配包含 X 或 Y 个 n 的序列的字符串。

n{X,} 匹配包含至少 X 个 n 的序列的字符串。

[abc] 查找方括号之间的任何字符。

[^abc] 查找任何不在方括号之间的字符。

(abc) 将abc当作一个单元进行匹配

(abc|123|456) |分割()内容单元

| 或者

^ ^ 开头

$ 结尾

(.\*?) 拒绝贪婪模式

转义字符

\w 数字字母下划线[a-zA-Z0-9\_]

\W 除了数字字母下划线[^a-zA-Z0-9\_]

\d 查找数字。 [0-9]

\D 查找非数字字符。[^0-9]

模式修正符

i 不区分大小写

g 全局匹配

二、鼠标事件

onclick 鼠标点击

onmouseover 鼠标移入

onmouseout 鼠标移出

onmousemove 鼠标移动

获取鼠标的坐标点

var ent = ent || window.event; 处理兼容性

var x = ent.clientX;

var y = ent.clientY;

获取元素的左边距 和 上边距

object.offsetLeft;

object.offsetTop;

=======================================

作业：

1. 代码敲3遍

2. 扩展：

使用正则 验证密码

第五天

任务清单：

1.html javascript -- 鼠标右击事件 -- oncontextmenu

2.html javascript -- 相册

3.html javascript -- 放大镜

4.html javascript -- 鼠标事件 -- div拖拽效果

5.html javascript -- 抽奖

6.html javascript -- 双击 --- 全选、全不选、反选

=============================================================

课堂笔记：

鼠标事件

鼠标单机 onclick

鼠标双击 ondblclick

鼠标右击 oncontextmenu

鼠标移入 onmouseover

鼠标移出 onmouseout

鼠标移动 onmousemove

鼠标按下 onmousedown

鼠标抬起 onmouseup

尺寸关键字

offsetLeft 左边距

offsetTop 上边距

offsetWidth 宽度

offsetHeight 高度

第六天

无

第七天

任务清单：

1.html javascript --- 事件 补充

2.html javascript --- 键盘事件

3.html javascript --- 事件冒泡

4.html javascript --- BOM --- 三种弹框

5.html javascript --- BOM --- 窗口操作

6.html javascript -- BOM -- location(浏览器url)对象

7.html javascript -- BOM -- navigator浏览器对象(获取浏览器相关信息)

8.html javascript -- BOM -- history对象(历史对象)

9.html javascript -- BOM --- screen对象(屏幕对象)

====================================

课堂笔记：

一、事件驱动

鼠标事件

鼠标单机 onclick

鼠标双击 ondblclick

鼠标右击 oncontextmenu

鼠标按下 onmousedown

鼠标抬起 onmouseup

鼠标移入 onmouseover

鼠标移出 onmouseout

鼠标移动 onmousemove

获取鼠标的位置和元素相关尺寸

鼠标坐标点：

ent.clientX;

ent.clientY;

元素的相关尺寸

offsetLeft 左边距

offsetTop 上边距

offsetWidth 自身的宽度

offsetHeight 自身的高度

键盘事件

键盘按下 onkeydown 获取ascii值 不区分大小写

键盘的ascii值

ent.keyCode

altKey 检测键盘上的alt是否按下

ctrlKey 检测键盘上的ctrl键是否按下

shiftKey 检测键盘上的shift键是否按下

键盘抬起 onkeyup

键盘按下并抬起 onkeypress 获取ascii值 区分大小写

表单事件

表单提交 onsubmit

表单重置 onreset

得到焦点 onfocus

失去焦点 onblur

值改变事件 onchange

值选中事件 onselect

框架事件

文档加载事件 onload

文档加载错误事件 onerror

文档内容滚动事件 onscroll

获取内容滚动的距离

document.docuemntElement.scrollLeft; 获取x轴滚动的距离

document.docuemntElement.scrollTop; 获取y轴滚动的距离

文档复制事件 oncopy

文档离开事件 onbeforeunload 针对IE浏览器 注意：返回的内容使用return

冒泡事件

事件冒泡： 就是这个事件发生之后，就要向这父级元素不断的扩散，这个过程就是事件冒泡

阻止事件冒泡关键字：

ent.cancelBubble = true; 默认为false

=====================================================

二、BOM

BOM是browser object model的缩写，简称浏览器对象模型

window 窗口对象

alert() 显示一个带有消息的警告框

confirm() 确认对话框

prompt() 问答对话框

close() 关闭浏览器对象

window.close()

open() 打开一个新窗口

window.open('1.html','','width=100,height=100,left=200,top=200;');

resizeTo() 修改窗口的大小

myWindow.resizeTo(400,400);

注意：谷歌只能用localhost访问

resizeBy() 增加窗口大小

myWindow.resizeBy(100,100);

注意：值可以为负值

moveTo() 将新窗口的位置移动到指定的位置

window.moveTo(x,y)

moveBy() 增加窗口位置

window.moveBy(x,y)

location 浏览器url对象

跳转

属性

window.location.href = '';

location.href = '';

location = '';

方法

location.replace('')

刷新

location.reload(true) //true 强制刷新

navigator浏览器对象 （了解）

appCodeName 返回浏览器的代码名

appName 返回浏览器的名称

appVersion 返回浏览器的平台和版本信息

cookieEnabled 返回指明浏览器中是否启用 cookie 的布尔值

platform 返回运行浏览器的操作系统平台

userAgent 返回由客户机发送服务器的 user-agent 头部的值

history对象

forward() 前进

back() 后退

go() 指定

screen 屏幕对象

width 获取屏幕的宽度

height 获取屏幕高度

availWidth 获取屏幕（不包含任务栏）的宽度

availHeight 获取屏幕（不包含任务栏）的高度

相关尺寸

documentElement.clientWidth 获取浏览器可视区域的宽度

documentElement.clientHeight 获取浏览器可视区域的高度

documentElement.scrollWidth 获取浏览器文档的全文宽

documentElement.scrollHeight 获取浏览器文档的全文高

documentElement.scrollTop 获取滚动条顶部的偏移量

documentElement.scrollLeft 获取滚动条左侧的偏移量

===============

作业：

代码敲1遍

抄2遍readme.txt

第八天

任务清单：

1.html javascript --- DOM --- 文档对象模型

2.html javascript -- DOM -- 元素查找

3.html javascript -- 购物车

4.html javascript -- 排除空白节点

5.html javascript -- 带有消息提醒的标题

6.html javascript -- 属性/类 操作

7.html javascript -- 电影院购票

============================================================

课上笔记

一、DOM是什么？

Document Object Model 文档对象模型 研究DOM，就是研究document对象

每个载入浏览器的 HTML 文档都会成为 Document 对象。

Document 对象使我们可以从脚本中对 HTML 页面中的所有元素进行访问。

HTML文档中的每个元素都是一个节点 , 无论是标签 标签属性 文本 注释甚至空白都算是一个节点

总结

标签节点 属性节点 文本节点

=========================================================

节点名称 大写的标签名 属性名 #text

节点类型 1 2 3

节点值 null 属性值 文本值

二、 节点获取

节点集合（元素集合）

获取所有的a连接 document.links;

获取所有的form表单 document.forms;

获取所有的img document.images;

通用方式

document.getElementById() 通过id属性来查找元素

document.getElementsByTagName() 通过标签名来查找元素

document.getElementsByClassName() 通过类名来查找元素

document.getElementsByName() 通过name属性查找元素

连贯获取

document.getElementById('box').getElementsByTagName('li');

关系查询

childNodes 获取所有的子节点

firstChild 获取第一个子节点

lastChild 获取最后一个子节点

parentNode 获取父节点

nextSibling 获取下一个兄弟节点

previousSibling 获取上一个兄弟节点

注意：空白也是一个节点

特殊

document.title 获取title

三、属性/类 操作

属性

attributes 获取所有的属性

getAttribute (key) 获取自定义的属性

setAttribute(key,val) 设置属性

removeAttribute(key) 移除属性

类

className = ''; 添加指定的类名

第九天

任务清单：

1.html javascript -- DOM -- 节点操作方法

2.html javascript -- ajax -- 原生

3.html javascript -- 封装ajax的使用

4.html javascript -- XML使用

===============================================

课堂笔记：

一、DOM 节点方法

创建节点

document.createElement('img');

在元素之后追加节点

box.appendChild(img);

在元素之前插入节点

box.insertBefore(newNode, oldNode);

替换一个节点

box.replaceChild(newNode, oldNode);

删除一个节点

box.removeChild(node);

克隆一个节点

img.cloneNode(true);

AJAX:

作用：

不刷新页面的前提下 和 服务器 交换 数据

数据格式：

字符串

json

xml

二、原生ajax

2.1 发送get

// 1. 实例化ajax对象

var xmlhttp = new XMLHttpRequest; //Request 请求

// 2. 监听

xmlhttp.onreadystatechange = function(){

// xmlhttp.status 获取当前http协议的状态码

// http 200 代表请求成功

// console.log(xmlhttp.status); ajax 状态码

/\*

0:请求未初始化

1:服务器连接已经建立

2:请求已经接收

3:服务器处理请求

4:服务器处理结束，返回结果

\*/

// console.log(xmlhttp.readyState);

if(xmlhttp.status == 200 && xmlhttp.readyState == 4){

// 接受服务器的返回结果

var res = xmlhttp.responseText; //接受文本

var res = xmlhttp.responseXML; //接受XML

alert(res);

}

}

// 3.配置

xmlhttp.open('get','./2.php?str=小鸡炖蘑菇',false);

// 4. 发送

xmlhttp.send();

2.2 发送post

//设置头信息

//使用POST发送，而且传递数据的话必须设置请求的头信息，发送的数据放在send方法的参数里

xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");

三、封装ajax使用

Ajax(recvType, bool) 返回 ajax对象

参数：

recvType 返回的数据类型：有三个值HTML(文本信息)、XML和JSON , 默认为HTML,传值时不区分大小写

bool 是一个boolean类型的值，true表示异步传输方式，false表示同步传输方式，默认为true

返回结果：对象

Ajax(recvType, bool).get(targetUrl,resultHandle)

get方法参数：

targetUrl:请求的URL地址

resultHandle:回调函数 执行成功时候调用的回调函数（当服务器返回结果执行的回调函数）

Ajax(recvType, bool).post(targetUrl, sendString, resultHandle)

post方法参数

targetUrl:请求的URL地址

sendString:发送的数据

resultHandle:回调函数

四、定义xml文件

注意：

1. 文档头 <?xml version='1.0' encoding='utf-8'?>

2. 标签 成对出现 有闭合

3. 属性必须加引号

4. 必须有根标签

作业：

基本 敲3遍

扩展：

验证用户名是否存在

第十天

一、ajax

同步 false

发送ajax之后的代码会等待 ajax的返回值

异步 true

发送ajax之后的代码不会等待 ajax的返回值

二、DOM table对象

insertRow() 添加tr对象

insertCell() 添加td对象

jquery

三、jquery

3.1 选择器

格式 $('选择符')

3.2 检测jquery引入

alert($)

3.3 文档事件

$(document).ready(function(){

})

简写：

$(function(){

})

3.4 注意：

原生的js方法 和 jquery方法不能混写

3.5 jqueyr选择器

基本

id

name

class

组合

层级选择器

A~B 之后所有的兄弟

A+B 紧挨着下一个兄弟

A B 后代选择器

A>B 子元素选择器

属性选择器

A[class] 匹配a元素包含attr属性

A[class=aa] 匹配a元素包含attr属性值等于val

A[class!=aa] 匹配a元素包含attr属性值不等于val

A[class^=aa] 匹配a元素包含attr属性值以val开头

A[class$=aa] 匹配a元素包含attr属性值以val结尾

A[attr\*=aa] 匹配a元素包含attr属性值包含val

A[.class][#id][...] 匹配a元素包含多个属性

伪基本选择器

获取第一个元素 :first

最后一个元素 :last

去除所有与给定选择器匹配的元素 ：not()

匹配所有索引值为偶数的元素，从 0 开始计数 :even

匹配所有索引值为奇数的元素，从 0 开始计数 :odd

匹配一个给定索引值的元素 eq(index)

匹配所有大于给定索引值的元素 gt(2)>2

匹配所有小于给定索引值的元素 lt(2)<2

第十一天

任务清单

1.html jquery -- 选择器 -- 子元素

2.html jquery -- 选择器 -- 表单

3.html jquery -- HTML代码/文本/值 --- 属性

4.html javascript -- css类操作

5.html jquery -- 动画

6.html jquery -- 自定义动画 -- animate

7.html jquery -- 全选/反选

8.html jquery -- 对象的访问

==================================================

课堂笔记

一、选择器

子元素

:nth-child(N) 匹配其父元素下的第N个子或奇偶元素

:first-child 匹配第一个子元素

:last-child 匹配最后一个子元素

:only-child 匹配父元素里面只有一个子元素

文本选择器

匹配包含给定文本的元素 contains(text)

匹配所有不包含子元素或者文本的空元素 empty

匹配含有选择器所匹配的元素的元素 has(selector)

匹配含有子元素或者文本的元素 parent

表单选择器

:checked 匹配所有选中的被选中元素

:selected 匹配所有选中的option元素

二、 属性

2.1 HTML代码/文本/值

html([val|fn]) 取得指定匹配元素的html内容。

text([val|fn]) 取得所有匹配元素的内容

val([val|fn|arr]) 获得匹配元素的当前值。

2.2 属性

attr() 获取（添加）属性

removeAttr() 移除属性

2.3 css类操作

addClass() 添加指定的类名

removeClass() 移除指定类名

toggleClass() 切换指定类名 如果这个类名存在(不存在)就是删除(添加)

三、动画

基本

show(time/ms,[callback]) 显示

hide(time/ms,[callback]) 隐藏

toggle(time/ms,[callback]) 切换显示隐藏

案例：

$('button:eq(0)').click(function(){

$('img:eq(0)').hide('slow',function(){

alert('memeda');

});

});

滑动

slideDown(time/ms,[callback]) 下滑动 显示

slideUp(time/ms,[callback]) 上滑动 隐藏

slideToggle(time/ms,[callback]) 切换 滑动

案例：

$('button:eq(4)').click(function(){

$('img:eq(1)').slideUp('5000');

});

自定义

animate({},time)

案例：

$('button:eq(0)').click(function(){

$('#aaa').animate({

"width":"300px",

"height":"300px",

"background-color":"#000",

"border":"2px solid #000"

},3000)

})

四、对象的访问

$.each() 对象遍历

index() 获取元素索引值

length 获取长度

size() 获取长度

第十二天

任务清单：

1.html jquery -- CSS操作

===================================================

课堂笔记：

一、CSS操作

css() 获取/设置css样式

css('attr')

css({'a1':'val1',..})

offset() 获取/设置 边距

offset()

offset({left:val,top:val})

event.pageX 获取鼠标x轴坐标点

event.pageY 获取鼠标y轴坐标点

scrollLeft() 获取内容x轴滚动距离

scrollTop() 获取内容y轴滚动距离

scrollTop(300);

position() 获取相对于父级元素定位距离

注意：有定位的父级元素

width() 获取元素自身的宽度

height() 获取元素自身的高度

二、事件

bind('事件驱动名称',fn)

live('事件驱动名称',fn)

live({}); //绑定多个事件

注意：bind 和 live 区别，live针对于后续追加的元素事件有效

one('事件驱动名称',fn) 触发一次事件

三、 效果

淡入淡出

fadeIn([speed],[easing],[fn])淡入

fadeOut([speed],[easing],[fn])淡出

fadeTo([[speed],opacity,[easing],[fn]])切换到指定的透明度

fadeToggle([speed,[easing],[fn]])// 淡入淡出的切换

基本

show(time/ms,[callback]) 显示

hiden(time/ms,[callback]) 隐藏

toggle(time/ms,[callback]) 切换显示隐藏

滑动

slideDown(time/ms,[callback]) 下滑动 显示

slideUp(time/ms,[callback]) 上滑动 隐藏

slideToggle(time/ms,[callback]) 切换 滑动

自定义

animate({},time)

stop() 停止

四、筛选

过滤

eq(index|-index) 查找第几个

first() 查找第一个

last() 查找最后一个

not()排除元素

has(expr|ele) 查找包含元素的

hasClass(class) 查找元素包含class 的 返回布尔值

第十三天

任务清单：

1.html jquery --- 筛选 -- 关系查找

2.html jquery -- 文档处理

3.html jquery -- 文档处理

4.html jquery -- 冒泡事件

5.html jquery -- ajax

6.html jquery -- 批量删除

7.html jquery -- 轮播

======================================================

课堂笔记

一、筛选

过滤

eq(index|-index) 查找第几个

first() 查找第一个

last() 查找最后一个

hasClass(class) 查找元素包含class 的 返回布尔值

has(expr|ele) 查找包含元素的

not排除元素

查找

children([expr])匹配所有的子级元素

find():查找后代元素

next():查找下一个

prev():查找上一个

parent():查找父级

siblings():查找所有的同辈元素(不包含自己)

二、 文档处理

内部处理

append() 后追加

prepend() 前置

外部处理

after() 在每个匹配的元素之后插入内容。

before() 在每个匹配的元素之前插入内容。

替换

replaceWith() 将匹配的元素替换成指定的html文档

删除

remove() 移除

empty() 清空

克隆

clone(true)

三、事件冒泡

原生： event.cancelBubble = true;

jquery: event.stopPropagation();

// 在js中 既能阻止事件冒泡 又能阻止元素默认行为

return false;

四、ajax

jQuery.get(url, [data], [callback], [type])

jQuery.post(url, [data], [callback], [type])

参数：

url 请求地址

data 数据 html json

callback 回调函数

type 返回的数据类型 xml html(默认) json

jQuery.ajax(url,[settings])

参数

$.ajax({

url:'请求地址',

type:'get(默认) post',

data:'发送到服务器的数据',

success:'请求成功后的回调函数',

dataType:'预期服务器返回的数据类型 html json xml',

async:true异步 false同步

});

作业：

敲2遍

抄2遍 jquery\_readme.txt

Jquery总结

一、选择器

1.1 基本

id

name

class

组合

1.2 层级选择器

A~B 之后所有的兄弟

A+B 紧挨着下一个兄弟

A B 后代选择器

A>B 子元素选择器

1.3 属性选择器

A[attr] 匹配a元素包含attr属性

A[attr=val] 匹配a元素包含attr属性值等于val

A[attr!=val] 匹配a元素包含attr属性值不等于val

A[attr^=val] 匹配a元素包含attr属性值以val开头

A[attr$=val] 匹配a元素包含attr属性值以val结尾

A[attr\*=val] 匹配a元素包含attr属性值包含val

A[attr1][attr2][...] 匹配a元素包含多个属性

1.4 伪基本选择器

获取第一个元素 :first

最后一个元素 :last

去除所有与给定选择器匹配的元素 ：not()

匹配所有索引值为偶数的元素，从 0 开始计数 :even

匹配所有索引值为奇数的元素，从 0 开始计数 :odd

匹配一个给定索引值的元素 eq(index)

匹配所有大于给定索引值的元素 gt(index)

匹配所有小于给定索引值的元素 lt(index)

1.5 子元素

:nth-child(N) 匹配其父元素下的第N个子或奇偶元素

:first-child 匹配第一个子元素

:last-child 匹配最后一个子元素

:only-child 匹配父元素里面只有一个子元素

1.6 文本选择器

匹配包含给定文本的元素 contains(text)

匹配所有不包含子元素或者文本的空元素 empty

匹配含有选择器所匹配的元素的元素 has(selector)

匹配含有子元素或者文本的元素 parent

1.7 表单选择器

:checked 匹配所有选中的被选中元素

:selected 匹配所有选中的option元素

二、属性

2.1 CSS

css() 获取/设置css样式

css('attr')

css({'a1':'val1',..})

offset() 获取/设置 边距

offset()

offset({left:val,top:val})

event.pageX 获取鼠标x轴坐标点

event.pageY 获取鼠标y轴坐标点

scrollLeft() 获取内容x轴滚动距离

scrollTop() 获取内容y轴滚动距离

scrollTop(300);

position() 获取相对于父级元素定位距离

注意：有定位的父级元素

width() 获取元素自身的宽度

height() 获取元素自身的高度

2.2 属性

HTML代码/文本/值

html([val|fn]) 取得指定匹配元素的html内容。

text([val|fn]) 取得所有匹配元素的内容

val([val|fn|arr]) 获得匹配元素的当前值。

属性

attr() 获取设置属性

removeAttr() 移除属性

css类操作

addClass() 添加指定的类名

removeClass() 移除指定类名

toggleClass() 切换指定类名 如果这个类名存在(不存在)就是删除(添加)

三、筛选

3.1 过滤

eq(index|-index) 查找第几个

first() 查找第一个

last() 查找最后一个

hasClass(class) 查找元素包含class 的 返回布尔值

has(expr|ele) 查找包含元素的

not排除元素

3.2 查找

children([expr])匹配所有的子级元素

find():查找后代元素

next():查找下一个

prev():查找上一个

parent():查找父级

siblings():查找所有的同辈元素(不包含自己)

四、 文档处理

内部处理

append() 后追加

prepend() 前置

外部处理

after() 在每个匹配的元素之后插入内容。

before() 在每个匹配的元素之前插入内容。

替换

replaceWith() 将匹配的元素替换成指定的html文档

删除

remove() 移除

empty() 清空

克隆

clone(true)

五、效果

淡入淡出

fadeIn([speed],[easing],[fn])淡入

fadeOut([speed],[easing],[fn])淡出

fadeTo([[speed],opacity,[easing],[fn]])切换到指定的透明度

fadeToggle([speed,[easing],[fn]])// 淡入淡出的切换

基本

show(time/ms,[callback]) 显示

hide(time/ms,[callback]) 隐藏

toggle(time/ms,[callback]) 切换显示隐藏

滑动

slideDown(time/ms,[callback]) 下滑动 显示

slideUp(time/ms,[callback]) 上滑动 隐藏

slideToggle(time/ms,[callback]) 切换 滑动

自定义

animate({},time)

stop() 停止

六、AJAX

$ ===> jQuery.get()

$.get(url, [data], [callback], [type])

$.post(url, [data], [callback], [type])

参数：

url 请求地址

data 数据 html json

callback 回调函数

type 返回的数据类型 xml html(默认) json

$.ajax(url,[settings])

参数

$.ajax({

url:'请求地址',

type:'get(默认) post',

data:'发送到服务器的数据',

success:'请求成功后的回调函数',

dataType:'预期服务器返回的数据类型 html json xml',

async:true异步 false同步

});

七、事件、对象访问

7.1 事件

bind('事件驱动名称',fn)

live('事件驱动名称',fn)

live({}); //绑定多个事件

注意：bind 和 live 区别，live针对于后续追加的元素事件有效

one('事件驱动名称',fn) 触发一次事件

7.2 对象访问

$.each() 对象遍历

index() 获取元素索引值

length 获取长度

size() 获取长度

八、事件冒泡

原生： event.cancelBubble = true;

jquery: event.stopPropagation();

// 在js中 既能阻止事件冒泡 又能阻止元素默认行为

return false;

Js原生创建ajax服务

// 1. 创建XMLHttpRequest()对象  
var xmlhttp = new XMLHttpRequest();  
// 2. 监听 onreadystatechange  
xmlhttp.onreadystatechange = function(){  
if(xmlhttp.readyState == 4 && xmlhttp.status == 200){  
var res = xmlhttp.responseText;   
alert(res);  
}  
}  
// 3. 配置open()  
xmlhttp.open('POST','3\_1.php',false);  
//设置头信息  
//使用POST发送，而且传递数据的话必须设置请求的头信息，发送的数据放在send方法的参数里  
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");  
// 4. 发送 send()  
xmlhttp.send('str=world');

js特效必须包括的三要素是：事件，事件处理程序，事件源