

JavaScript及Web网页前端开发

第4讲 JavaScript基础

唐大仕

课程进度

12

综合应用

13

复习

10

web api

11

ajax/服务端

9

Bootstrap

8

Jquery

5

HTML5

6

CSS3

7

Javascript进阶

2

HTML

3

CSS

4

Javascript基础

1 序

如何学习

■ 文档

- <https://www.w3school.com.cn/jsref/index.asp> 参考文档
- <https://www.w3school.com.cn/js/index.asp> 教程
- <https://developer.mozilla.org/zh-CN/docs/Web/JavaScript>

■ 多练

- 浏览器中F12
- 进入到console

■ 参考书 《疯狂HTML 5/CSS3/JavaScript讲义》

在线工具

- <http://runjs.cn/code> 中文
- <http://jsdo.it> 中文
- <http://www.w3school.com.cn/tiy/t.asp> 中文
- <http://jsfiddle.net/> 4区, 可调框架
- 重要 <http://codepen.io>
- 重要 <https://c9.io/>
- <http://www.playmycode.com/> 可实时
- <http://www.weebly.com> 可视化

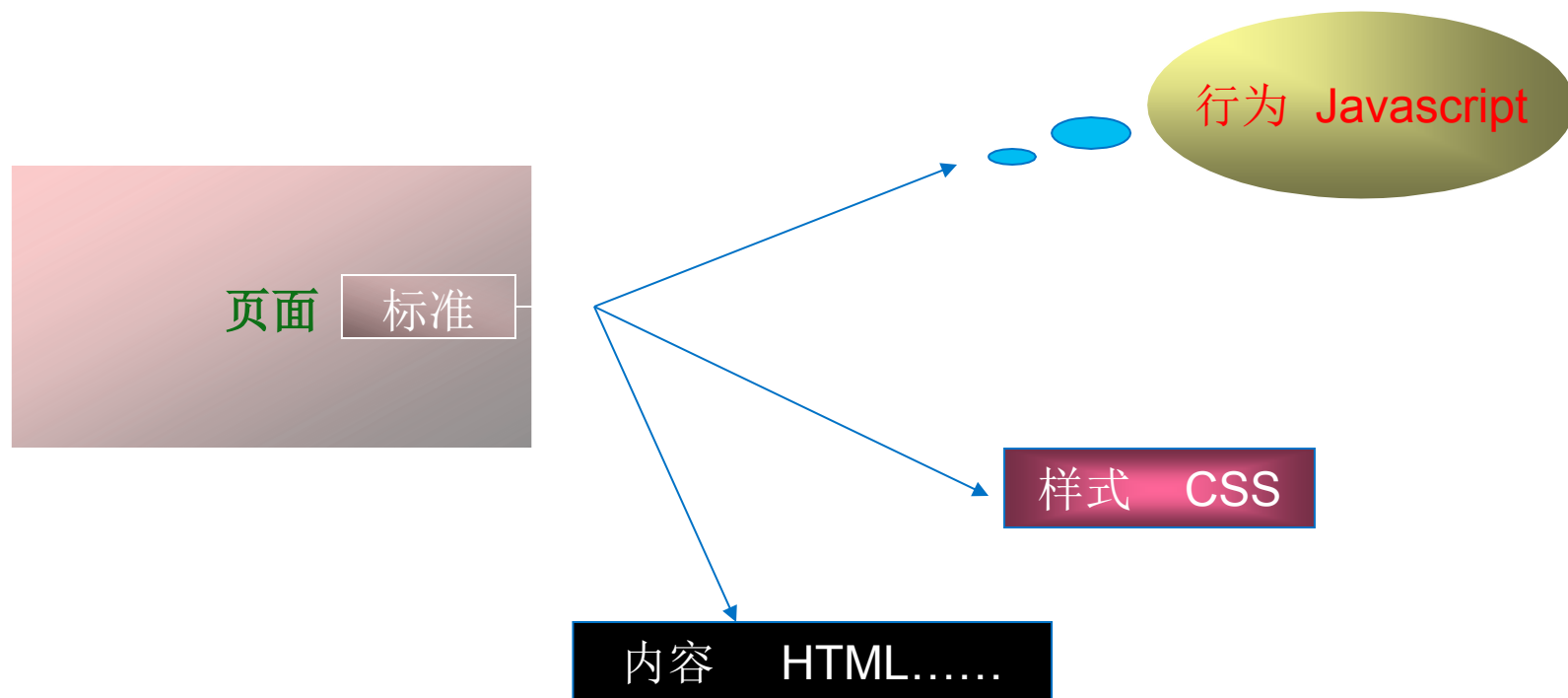
本讲内容

- JS简介
- JS语言
- JS访问HTML

JS简介

了解JavaScript

- 是一种基于对象和事件驱动并具有安全性的**解释性**语言，其目的就是增强Web客户交互。弥补了HTML的缺陷。



JavaScript简介

■ JavaScript与Java两种完全不同的语言

- Java是SUN公司推出的新一代面向对象的程序设计语言，特别适合Internet应用程序开发，前生是Oak语言。
- JavaScript是Netscape公司的产品，是一种可嵌入到WEB当中的基于对象和事件驱动的解释性语言，前生是LiveScript

■ JavaScript的正式名称叫 ECMAScript

- ECMA国际（前身为欧洲计算机制造商协会,英文名称是European Computer Manufacturers Association）
- 现在多数浏览器支持ES5
- ES6（ECMAScript 2015）在2021年已得到大多数浏览器支持

JavaScript与Java不同点

JavaScript	Java	
运行于网页中	运行于桌面或后台	
解释	编译	
弱类型变量	强类型变量	
基于对象	面向对象	

JavaScript可以做什么

■ 交互性

- 使网页更具有交互性，给用户提供更好的、更令人兴奋的体验。

■ 有效性

- 确保用户在表单中输入有效的信息，可以节省你的业务开支。

■ 动态性

- 即时创建HTML页面。

■ 应用性

- 可以处理表单，设置cookie，创建基于Web的应用程序。

JavaScript不可以做什么

■ 主要基于安全的考虑

- 不允许读写客户机器上的文件。
- 不允许写服务器上的文件。
- 不能关闭不是由它打开的窗口。
- 不能从来自另一个服务器的已经打开的网页中读取信息。

脚本写在哪儿？

- 写到 `<script> </script>` 中
- 或者写到一个单独的文件
 - 然后引用它 `<script src="xxxx.js"> </script>`

`src=...`

HTML 文档

```
<HTML>
  <HEAD>
    .....
  </HEAD>
  <BODY>
    .....
  </BODY>
</HTML>
```

```
<SCRIPT >
  function HELLO()
  {
    .....
  }
</SCRIPT>
```



第一个JavaScript程序

■ 示例：见 0-simple

```
<html>
```

```
<body>
```

```
<script >
```

```
    document.write("Hello World!");
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript快速入门

■ JS书写

- `<script>... </script>`

■ 单词、语句、段落

- 单词：var 区分大小写

- 语句：带分号（可不带）

- 段落：function xxx() { ... }

■ 与HTML交互

- 元素：获得DOM对象 `document.getElementById('..')`

- 对象：属性、方法、事件

- 事件：onclick onmouseover

- 周期执行：`setInterval(fun, 100);`

JS语言

JS的基础语法

- JS的基础语法类似于C、C++、C#、Java
- JavaScript区分大小写
 - 一般小写
 - 如果是多个单词，一般用驼峰(Camel)式
 - getElementById
- 语句后面用分号；但可以省略分号
- 多行注释 `/* ...*/` 单行注释 `//`
 - vscode中用Ctrl+/
 - 回顾 html注释 `<!-- -->` css注释 `/* */`

单词：变量与常量

■ var表示一个变量(variable)

- `var a=1+2;` (有作用域提升问题)

- `let x=9;`(局部变量) `const y =10;` (常量)

■ 它是弱类型的，或者说是动态类型的

- `a=4` 是数值类型

- 然后 `a="aaa"`就成了字符串类型

■ 示例

- 见 1-var

JS数据类型

■ 基本数据类型

- ❑ 数值 (Number) : 整数和实数 (内部都用实数) 如100 , -1.6, 2.7e-5
- ❑ 字符串型 (String) : 用 `""` 号或 `''` 括起来的字符 (内部用UTF-16) ``
- ❑ 布尔型 (Boolean) true或false
- ❑ undefined
- ❑ null

■ 引用数据类型

- ❑ 对象(Object), 如document
- ❑ 数组(Array), 如 `a = [1,2,3,4]` ; 数组有长度属性`a.length`
- ❑ 函数(Function), 如 `function f(){....}`

数据类型转换

- 由于它是弱类型，有时需要强制处理类型

- 常用的方法

- `parseInt("124px", 10)` 强制当成10进制整数

- `parseInt("123.67")` 强制当成整数(124)

- `parseFloat(xxx)` 强制当成实数

- `(3.14).toString()` 强制转成字符串

- `isNaN("123xy")` 判断是否不是数值

- `!! a` 强制转成boolean类型

运算符

■与C/Java类似

- 判断闰年 `y % 4 == 0 && y % 100 != 0 || y % 400 == 0`

- `a = a || "100px";` 特殊用法 (短路)

- `fun && fun()`

- `!a && cnt++`

■字符串连接，使用+

- 如 `a=100, obj.style.left = a + "px";`

语句

■ 语句后一般写分号

- 也可以不写分号

■ 分支语句

- `if(条件){...} else {...}`
- `switch`语句
- 也有“非空即真”的用法

■ 循环语句

- `for(let i=0; i<10; i++){document.write(i);}`
- `while(条件){.....}`

示例

■ 见 2-statements

- 运动的小球
- 自动出题并判分

段落：函数

■ 将一段程序写到一个函数中

- 定义函数 `function xxxx() { }`

- 然后调用之，`xxxx()`

- 注意函数的定义与函数的调用

■ 函数可以带参数，也可以有返回值

- `function add(a, b){ return a+b; }`

■ 匿名函数

- `function (a,b){ return a+b;}` `(function (){...let a=5;...})();`

■ Lambda表达式 (新 es6)

- `(a, b)=>a+b` `x=>{console.log(x);}`

示例

■ 见3-function

□ 求和

- 体会"任务分解"

□ 附加html元素

- 体会"代码重用"

JS访问HTML

简单的输入输出

■ 可以用

- `prompt()` 提示性输入 `window.prompt()`
- `alert()` 弹出信息 `window.alert()`
- `document.write()` 输出到文档中
- `console.log()` 表示控制台输出

更一般的输入输出

■使用html元素

- `<input>` `document.getElementById('xxx').value`
- `<div>` `document.getElementById('xxx').innerHTML`
 - `<div id="xxx"> sdfjlasdfjasdkjf</div>`
- 另外，样式、图片、媒体也可以表示输出

JS操作页面对象

■ JS两个重要特点

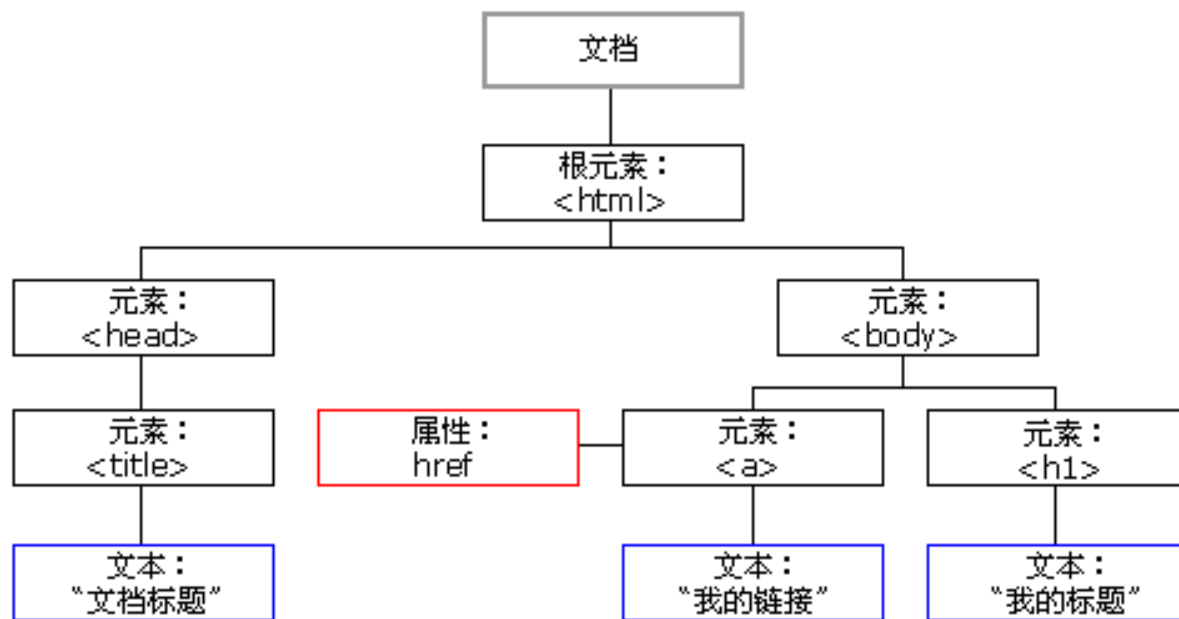
- 基于对象、事件驱动

■ 常用操作

- 1. 文档对象
- 2. 事件
- 3. 改变对象的内容
- 4. 改变对象的css
- 5. 定时执行动作

文档对象

■ DOM: Document Object Model



获得对象

■ document.getElementById("xxx")

➤ 也可以直接用id表示对象，如 `xxx.innerHTML = 'hello'`；不推荐

■ 其它方法

- document.getElementsByTagName

- document.getElementsByName

- document.getElementsByClassName() 等

- 得到的是数组，可以用[0]来取得首个元素, .length取得个数

■ 更一般地用css来查询

- document.querySelector(".content .news") 有人称之为QSA

- document.querySelector(".news")

DOM对象的三要素

■ 属性(attribute, property)

- `var myhref=document.getElementById("mylink").href;`

- ``

- 一般是名词或形容词，表示对象的特征或状态，本质是个变量

■ 方法(method)

- `document.write(txt);`

- 一般是动词，表示对象的行为或功能，本质是个函数

■ 事件(event)

- `<button onclick="xxxx">点击</button>`

- 一般是on+动词，表示对象与外界的交互或对象的变化，本质是个消息通知

方法论：事件驱动

■ 例：

- `<div id="msg">hi</div>`

- `<button onclick="msg.innerHTML='hello'">click me</button>`

■ 什么对象、发生什么事件、做什么事

事件的几种写法

■ (1) 直接写到元素上

□ `<button onclick="语句 或 函数名 ()">点击</button>`

■ (2) 在script中写

- `document.getElementById("xxx").onclick=函数名` 不能写圆括号

- 或者直接写成匿名函数

- `document.getElementById("xxx").onclick=function () {.....}`

- ES6 中匿名函数可写成Lambda表达式 (箭头函数) `(x)=>{.....}`

□ (3) 更一般的写法 (新)

- `xxx.addEventListener("click", function(e){.....})` 这里没有on

常见事件

■ body的onload

- 多用于初始时要做的事情

■ 一般对象的事件（ 注意是全小写 ）

- onclick
- onmousedown onmouseup
- onmouseover onmouseout
- onchange onselect onfocus onblur
- onload onunload onkeypress

■ 示例：见4-events

改变对象的属性

- 如改变链接

- .href

- 如改变文本框

- .value

改变对象的内容

■.innerHTML

- 这是最常用的方法，可以包含html标记

- `<div id="aa"></div>` `<button onclick=`

- `'document.getElementById("aa").innerHTML=new Date();'>点击显示时间</button>`

- 除了innerHTML，还有.innerHTMLText 或 .textContent，它们只能是文本

■还可以用（新）

- `d = document.createElement('div')` 及 `.createTextNode("文本")`

- `xxx.appendChild(d)`

改变对象的样式

■ 改变具体的样式 (注意是驼峰式大小写)

- `oDiv.style.cssText = "width:300px";`
- `oDiv.style.width = "300px";`
- `oDiv.style.backgroundColor = "yellow";`

■ 改变样式名或样式内容

- `oDiv.className = "onebox round";`
- `oDiv.setAttribute('style', 'width:300px; color:red;');`

■ 添加或删除样式(新)

- `oDiv.classList.add("news")`
- `oDiv.classList.remove("news")`

周期执行任务

■ `setInterval(myfun, 1000);`

■ `var i=10;`

■ `function myfun(){`

- `document.getElementById('mydiv').innerHTML=new Date();`

- `document.getElementById('mydiv').style.left= i+ "px";`

- `i+=10;`

■ `}`

■ 或用 `setTimeout(myfun, 1000)`

- 并在 `myfun` 中自己再调用 `setTimeout`

示例

■ 见5-dom

□ 雪花飘飘

小结

- 1.文档对象 `document.getElementById`
 - 2.事件 `.onclick`
 - 3.改变对象的内容 `.innerHTML`
 - 4.改变对象的样式 `.style.xxxx`
 - 5.定时执行动作 `setInterval(fun, 100)`
-
- 基于对象（三要素：属性、方法、事件 查mdn文档）
 - 方法论：事件驱动

