

JavaScript语言

JSD WEB jAVASCRIPT



目录 | CONTENTS

- 01 JavaScript概述
- 02 基础语法
- 03 BOM和DOM
- 04 MVC和MVVM设计模式

01

JavaScript概述

什么是JavaScript

- JavaScript（简称“JS”）由Netscape公司在1995年首次推出，最初被称为 Mocha，然后改名为 LiveScript，最终被命名为 JavaScript。目前已经成为最受欢迎的编程语言之一。
- JavaScript通常用于Web页面中，通过HTML和CSS来实现交互效果和动态内容。
- JavaScript和Java虽然名称相似，但实际上是两种完全不同的编程语言。
- **作用: 负责给页面添加动态效果**



- 动态类型：JavaScript是一种弱类型的语言，不需要指定变量类型，变量的类型在运行时根据值自动推断。
- 解释性：JavaScript是一种解释型语言，不需要编译就可以直接在浏览器中运行。
- 基于对象。JavaScript是一种基于对象的脚本语言，它不仅可以创建对象，也能使用现有的对象。

一门编程语言包括哪些内容?

- 变量
- 数据类型
- 运算符
- 各种语句
- 方法
- 面向对象

02

语言基础

- JavaScript语言属于弱类型语言,声明变量不需要指定类型
- JavaScript语言中通过let或var声明变量
 - ✓ let声明的变量, 作用域和Java语言类似,块级作用域(推荐使用)
 - ✓ var声明的变量, 全局作用域

基本数据类型

- 字符串 (String)、数字 (Number)、布尔 (Boolean)、未定义 Undefined

引用类型

- 对象 (Object)、数组 (Array)、函数 (Function)

获取变量类型的方式: `typeof 变量名;`

- 算术运算符: + - * / %
 - ✓ 除法运算会自动根据结果转换整数或小数
 - java : `int x = 5; int y=2; int z = x/y; z=2`
 - js : `let x = 5; let y=2; let z = x/y; z=2.5 x=6 z=3`
- 关系运算符: > < >= <= != ==和===
 - ✓ ==: 先统一等号两边变量的类型, 再比较值 `"666"==666 true`
 - ✓ ===: 先比较类型, 类型相同后再比较值 `"666"===666 false`
- 逻辑运算符: && || ! 只支持短路与和短路或
- 赋值运算符: = += -= *= /= %=
- 三目运算符: 条件?值1:值2

- 分支语句:if和else
- for 新循环 for(Person p : persons) JS: for(let p of persons)
- while
- switch case

如何在HTML页面中引入JavaScript代码

- 引入方式:
 - ✓ 内联:在标签的事件属性中添加JS代码,当事件触发时执行.
 - 事件:指系统提供的一系列时间点.
 - 点击事件:当用户点击元素的时间点.
 - ✓ 内部:在页面的任意位置可以添加script标签,在标签体内写js代码,建议写在body结束标签的附近
 - ✓ 外部:在单独的js文件中写js代码,在html页面中通过script标签的src属性引入,如果script标签引入了js文件则不能在标签体内继续写js代码

- Java语言的方法和JavaScript的方法对比
 - ✓ Java: public 返回值 方法名(参数列表){方法体}
 - ✓ JS: function 方法名(参数列表){方法体}
- 三种定义方法的方式:
 - ✓ function 方法名(参数列表){方法体}
 - ✓ let 方法名 = function(参数列表){方法体}
 - ✓ let 方法名 = new Function("参数1","参数2","方法体");

- 通过选择器获取页面中的元素对象
 - ✓ `let 元素对象 = document.querySelector("选择器");`
- 获取和修改控件的值
 - ✓ `input.value` 获取
 - ✓ `input.value="xxx"` 修改
- 获取和修改元素的文本内容 例:`<div>文本内容</div>`
 - ✓ `变量.innerHTML` 获取
 - ✓ `变量.innerHTML = "xxx"` 修改

计算器练习

课堂练习

5

2

加

减

乘

除

结果:7



猜数字练习

50

猜一猜

猜小了!



03

BOM和DOM

什么是BOM?

BOM (Browser Object Model) 是指浏览器对象模型，它是JavaScript与浏览器之间进行交互的接口。BOM 提供了一组对象，这些对象代表着浏览器相关内容，并提供了访问和控制这些对象的方法。以下是一些常见的 BOM 对象：

- window
- location
- history

- window对象是JavaScript中表示当前浏览器窗口的内置全局对象。它包含了浏览器窗口的所有属性和方法，可以通过该对象来访问和控制当前窗口的各种行为。
- 常用方法:
 - ✓ isNaN(变量) 判断变量是否是NaN
 - ✓ parseInt() 将字符串或小数转成整数
 - ✓ parseFloat() 将字符串转成小数
 - ✓ alert("xxx") 弹出提示框
 - ✓ confirm("xxx")弹出确认框
 - ✓ prompt("xxx")弹出文本框

- location.href 获取或修改浏览器的请求地址
- location.reload() 刷新页面

- history.length 历史页面的数量
- history.back() 返回上一页面 后退
- history.forward() 前往下一页面 前进
- history.go(n) n正值代表前进 n负值代表后退 0代表刷新

什么是DOM?

Document Object Model 文档对象模型, 包含和页面相关的内容

- 通过选择器获取页面中的元素对象
 - ✓ `let 元素对象 = document.querySelector("选择器");`
- 获取和修改控件的值 `元素对象.value`
- 获取和修改元素的文本内容 `元素对象.innerText`
- 创建元素对象
 - ✓ `let d = document.createElement("div");`
- 添加元素到某个元素里面
 - ✓ `元素对象.append(新元素);`

员工列表练习

员工列表

姓名	工资	工作
刘德华	3000	销售
张学友	5000	人事
郭富城	8000	程序员



综合练习

个人信息

照片:	
姓名:	传奇哥
年龄:	18
好友:	<ul style="list-style-type: none"> • 貂蝉 • 西施 • 武则天

加载数据



- MVC设计模式就是将实现前端业务的所有代码划分为三部分
 - ✓ M: model 模型, 对应数据相关代码
 - ✓ V: View 视图, 对应的是页面标签相关
 - ✓ C: Controller 控制器, 对应的是将数据显示到页面中的过程代码
- 前端MVC设计模式存在弊端: 在Controller控制器部分 需要频繁的进行DOM相关操作(遍历查找元素), 比较浪费资源, MVVM设计模式可以解决此问题

MVVM设计模式也是将实现前端业务的所有代码划分为三部分

- M: model 模型, 对应数据相关代码
- V: View 视图, 对应的是页面标签相关
- VM:视图模型, 视图模型负责将页面中可能发生改变的元素和某个变量在内存中进行绑定, 并对变量进行监听, 当变量发生改变时, 会从内存中找到和变量所对应的元素, 让元素进行改动. 这样就不用像MVC设计模式中通过遍历查找的方式查找元素了, 从而提高执行的效率。