Java 语言

变量

数据类型

运算符

条件结构

循环结构

数组

面向对象 类 对象

属性 方法

JS JavaScript 脚本语言 解析性

应用场景：在网页上应用 HTML CSS

① 核心语法：实现基础业务逻辑

② DOM 对象：通过js 操作HTML 内容和元素

③ BOM 对象：通过js 操作浏览器实现部分功能

# 一、核心语法

## 1、应用场景和角色

① HTML 设计网页内容

② CSS 设计网页内容样式

③ JS 实现网页动态效果

## 2、输出操作

① 通过 ***window***.document.getElementById(**"body"**).innerHTML=**"";**

|  |
| --- |
| <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>JS核心语法</**title**> </**head**> <**body id="body"**> </**body**> *<!--在网页文档中输出网页内容--> <!--  内容和元素 - 网页文档中 - 浏览器  浏览器窗口 - 网页文档对象 - 元素 | 内容 | 样式 -->* <**script**>  ***window***.document.getElementById(**"body"**).innerHTML=**"<h1>js输出网页内容</h1>"**; </**script**> </**html**> |

② 弹出提示框 ***window***.alert(**"早上好，同学们。"**);

|  |
| --- |
| <**body id="body"**> </**body**> *<!--在当前窗口中弹出提示框--> <!--  浏览器窗口 - 浏览器对象 功能 -->* <**script**>  ***document***.getElementById(**"body"**).innerHTML=**"<h1>js输出网页内容</h1>"**;  ***window***.alert(**"早上好，同学们。"**); </**script**> |

③ 弹出输入框 ***window***.prompt(**"请输入姓名？"**,**"默认值"**);

|  |
| --- |
| <**body id="body"**> </**body**> *<!--在当前窗口中弹出提示框--> <!--浏览器窗口 - 浏览器对象 功能-->* <**script**>  ***document***.getElementById(**"body"**).innerHTML=**"<h1>js输出网页内容</h1>"**;  ***window***.prompt(**"请输入姓名？"**,**"默认值"**); </**script**> |

④ 弹出带有确定和取消按钮的提示框*confirm*(**"是否确认删除？"**);

|  |
| --- |
| <**body id="body"**> </**body**> *<!--在当前窗口中弹出提示框--> <!--浏览器窗口 - 浏览器对象 功能-->* <**script**>  ***document***.getElementById(**"body"**).innerHTML=**"<h1>js输出网页内容</h1>"**;  *confirm*(**"是否确认删除？"**); </**script**> |

## 3、语法规范

① 在编写js脚本时，必须在用 英文格式分号来结尾

② 顺序，网页内容有浏览器逐行解析的，所以要注意加载的顺序

③ 标识符的要求，不能用数字开头、$、字母、驼峰命名

④ 大小写是敏感的，区分大小写

## 4、注释

① 单行注释： // 注释内容

② 多行注释：/\* 注释内容 \*/

## 5、变量

① 变量的声明

|  |
| --- |
| <**script**>  */\*  js中的变量的数据类型是由变量的实际 的值来动态确定的  声明变量，使用：var 变量名;  \*/* **var *name*** ; </**script**> |

② 变量的赋值

|  |
| --- |
| <**script**>  */\*  js中的变量的数据类型是由变量的实际 的值来动态确定的  声明变量，使用：var 变量名;*  *直接使用 = 赋值  \*/* **var *name*** = **"小徐"**;  *alert*(***name***); </**script**> |

## 6、数据类型

① 字符串 String

② 数值 Number

③ 布尔 Boolean

④ 数组

⑤ 对象 Object

⑥ null

⑦ undefined 未定义 变量只生命 未赋值或不存在的变量

判断数据类型的 方法：typeOf()

|  |
| --- |
| <**script**>  */\*  js中的变量的数据类型是由变量的实际 的值来动态确定的  声明变量，使用：var 变量名;  \*/* **var *name*** = **"小徐"**; *// string 类型* **var *age*** = 28.55; *// number* **var *flag*** = **true**; *// boolean* **var *array*** = [1,2,3,4,5,6,7,8]; *// object* **var *person*** = {  **"name"**:**"小徐"**,  **"age"**:28,  **"sex"**:**"男"** }; *// object* **var *student*** = **null**; *// object* **var *undefineds*** ; *// undefined 变量声明后未赋值 或者 该变量不存在  alert*(**typeof** undif); </**script**> |

④ 数组类型

|  |
| --- |
| <**script**>  */\*  js中的数组，初始化分为静态初始化和动态初始化  实际操作数据数据的过程中，必须要使用 下标 来进行操作  可以通过 length 属性获取数组中元素的个数  \*/  /\*① 动态初始化\*/* **var *array*** = **new *Array***(); *// object* ***array***[0] = **"name"**;  ***array***[1] = **"age"**;  ***array***[2] = 30;  */\*② 静态初始化\*/* **var *array2*** = **new *Array***(**"1"**,**"小徐"**,**"男"**,55);  **var *array3*** = [**"2"**,**"小丽"**,**"女"**,20];  *alert*(***array3***.length); </**script**> |

⑤ 对象类型

|  |
| --- |
| <**script**>  */\*  js中的对象：  \*/  /\*var person = new Object();  person.name = "小徐";  person.age = 28;\*/* **var *person*** = {  **name**:**"小徐"**,  **age**:28,  **sex**:**"男"**,  toString:**function** () {  **var** str = **"姓名："**+**this**.**name**+**",年龄："**+**this**.**age**+**",性别："**+**this**.**sex**;  **return** str;  }  };  *alert*(***person***.toString()); </**script**> |

## 7、数据类型转换

① 数值转字符串

|  |
| --- |
| <**script**>  */\*js需要解决的问题， 将数值类型的数据转换成字符串类型  ① 在数值的基础上拼接一个空字符  ② 使用String方法  ③ 调用Number对象中toString()  \*/* **var *num1*** = (28).toString();  **var *num2*** = ***String***(30);  **var *num3*** = 50+**""**;  *alert*(**"两个数拼接之后的值："**+(***num1***+***num2***+***num3***)); </**script**> |

② 字符串转数值

|  |
| --- |
| <**script**>  */\*  js 实现计算两个数的和  需要解决的问题， 将字符串数字 转换成数值类型的  需要使用 Number()方法  \*/  /\*接收用户输入的数据  问题：由prompt()方法接收到的用户输入的数据是字符串类型  需要对接收到的两个数据进行 数据类型转换，将数据转换成数值类型  转换成数值类型：Number(需要转换的数据);  \*/* **var *firstNum*** = ***Number***(*prompt*(**"请输入第一个数值？"**));  **var *secondNum*** = ***Number***(*prompt*(**"请输入第二个数值？"**));  *alert*(**"两个数值相加之后的和为："**+(***firstNum***+***secondNum***)); </**script**> |

## 8、运算符

① 算数运算符

加+、减-、乘\*、除/、取余(取模)%、自增++、自减—

|  |
| --- |
| <**script**>  */\* 自增和自减  ① 必须要用在已经初始化的变量上  ② 变量的要求，值是数字，可以是字符串类型或数值类型  \*/* **var *num*** = **"53"**;  *alert*(++***num***); </**script**> |

② 逻辑运算符

与&&、 或||、 非！

③ 比较元素符

相等==、全等===、不等!=、大于等于>=、小于等于<=、大于>、小于<

|  |
| --- |
| <**script**>  */\*  比较运算符  相等 == ：判断两个表达式是否相等时，不区分数据类型  如果说需要判断两个表达式的值和数据类型都相同  全相等 === ：判断两个表达式的值和数据类型都相等  全不等 !== :判断两个表达式的值和数据类型都不相同  \*/* **var *num1*** = 50;  **var *num2*** = **"50"**;  *alert*(***num1***===***num2***);  *alert*(***num1***!==***num2***); </**script**> |

④ 赋值运算符

=、+=、-=、\*=、/=、%=

## 9、条件结构

① if 、 if else 、 if else if

② switch case break default

## 10、循环结构

① for

② while

③ do while

## 11、函数（类似于java中的方法）

① 函数的执行 是由 事件驱动 或 被直接调用 的可重复执行的代码块

② 定义函数 需要通过 function 函数名(参数列表){ //代码块 return 返回值 }

③ 调用函数时需要注意其参数的顺序和 参数的数据类型

④ 调用函数时可以设置返回值，返回值通过return关键字进行返回

⑤ 定义变量时，如果涉及到了函数，需要考虑定义的是全局变量还是局部变量

## 12、日期对象

|  |
| --- |
| <**script**>  */\*  Date对象  想要获取当前的时间  var date = new Date();  \*/* **function** *getDateStr*() {  **var** date = **new *Date***();  */\*需要的日期信息： 年份 月分 天 小时 分钟 秒\*/* **var** year = date.getFullYear(); *//年份* **var** month = date.getMonth()+1;*//月份* **var** day = date.getDate(); *//天* **var** hour = date.getHours(); *//小时* **var** minutes = date.getMinutes(); *// 分钟* **var** second = date.getSeconds(); *//秒* **if**(day<10){  day = **"0"**+day;  }  **var** dateTimeStr = year+**"-"**+month+**"-"**+day+**" "**+hour+**":"**+minutes+**":"**+second;  **return** dateTimeStr;  }  *alert*(*getDateStr*()); </**script**> |