

JavaScript

一、Javascript介绍

JavaScript（简称“JS”）是当前最流行、应用最广泛的客户端脚本语言，用来在网页中添加一些动态效果与交互功能，在 Web 开发领域有着举足轻重的地位。

1. JavaScript 与 HTML 和 CSS 共同构成了我们所看到的网页

- HTML 用来定义网页的内容，例如标题、正文、图像等
- CSS 用来控制网页的外观，例如颜色、字体、背景等
- JavaScript 用来实时更新网页中的内容，例如从服务器获取数据并更新到网页中，修改某些标签的样式或其中的内容等，可以让网页更加生动。

二、JavaScript使用方法

1. 在body内部写script标签、在标签内写js代码

```
<body>
  <script>
    // 写js逻辑代码
  </script>
</body>
```

2. 在head头部标签中 通过script引入js文件

```
<head>
  <!-- 网页采用utf-8的编码格式 -->
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- 网页标题 -->
  <title>Document</title>
  <!-- 引入js文件 -->
  <script src="./index.js"></script>
</head>
```

三、JavaScript输出

1. **console.log** 在控制台输出（一般在程序中用来调试代码，打印日志）

注意：必须打开网页控制台才能看到输出的内容

```
<body>
  <script>
    console.log('hello JavaScript!');
  </script>
</body>
```

2. 在网页输出（一般在程序中用来调试代码，提示报错）

注意：打开网页自动弹出

```
<body>
  <script>
    alert('hello JavaScript!')
  </script>
</body>
```

拓展：使用 `prompt()` 函数：`prompt()` 函数会弹出一个对话框，提示用户输入信息，并返回用户输入的字符串。

```
<body>
  <script>
    let name = prompt("请输入您的姓名：");
    console.log("您输入的姓名是：" + name);
  </script>
</body>
```

四、JavaScript注释

- 单行注释

```
<body>
  <script>
    // 这是注释内容
    alert('hello JavaScript!')
  </script>
</body>
```

- 多行注释

```
<body>
  <script>
    /* 这是注释内容
    很多注释 */
    alert('hello JavaScript!')
  </script>
</body>
```

补充：

```
<!-- html注释 -->
/* css注释（单行多行通用） */
// js注释
/* js多行注释*/
```

五、变量

- 变量声明

```
// 声明变量
let 小皮;
```

- 变量赋值

```
<script>
  // 声明变量
  let 小皮;
  // 变量赋值
  小皮 = '我家的一条狗'
</script>
```

- 变量声明并赋值

```
<body>
  <script>
    let 小皮 = '我家的一条狗'
    console.log(小皮);
  </script>
</body>
```

六、JavaScript的数据类型

6.1 基本数据类型

- 字符串型 (String)
- 数值型 (Number)
- 布尔 (Boolean)
- undefined (Undefined)
- null (Null)

6.2 String 字符串

String用于表示一个字符序列，即字符串。字符串需要使用 **单引号** 或 **双引号** 括起来。

```
<script>
  let a = '我是一个字符串a'
  let c = "我是一个字符串b"
</script>
```

6.3 Number 数值型

Number 类型用来表示**整数**和**浮点数**，最常用的功能就是用来表示10进制的整数和浮点数。

```
<script>
  // 整数
  let a = 10
  // 浮点数
  let c = 10.25
</script>
```

6.4 Boolean

布尔型也被称为逻辑值类型或者真假值类型。

```
<script>
  // true 代表真
  let a = true
  // false 代表假
  let c = false
</script>
```

6.5 Undefined

在使用 let 声明变量但未对其加以初始化时，这个变量的值就是 undefined。

```
<script>
  // 变量声明 没有赋值 就会打印undefined
  let a;
  console.log(a);
</script>
```

6.6 Null

空，一般用来初始化

```
<script>
  // 当变量需要为空的时候 就可以写null
  let a = null
  let b = null
  // js中null 类似于 Python中的None
</script>
```

6.7 typeof 判断类型

场景：基本数据类型的判断

```
<script>
  let a = '我是字符串类型'
  let b = 10
  let c = undefined
  let d = null
  let e = true
  console.log(typeof a); // string
  console.log(typeof b); // number
  console.log(typeof c); // undefined
  console.log(typeof d); // 如果值是null 打印会为object类型
  console.log(typeof e); // boolean
</script>
```

七、运算符

7.1 算术运算符

用于表达式计算。

1. 加法 +
2. 减法 -
3. 乘法 *
4. 除法 /
5. 自增 ++
6. 自减 --

案例：

```
<script>
  let a = 30
  let b = 20
  console.log(a+b); // 加法 结果：50
  console.log(a-b); // 减法 结果：10
  console.log(a*b); // 乘法 结果：600
  console.log(a/b); // 除法 结果：1.5
</script>
```

```
console.log(++a); // 自增 结果：31
```

```
console.log(--a); // 自减 结果：29
```

7.2 赋值运算符

赋值运算符用于给 JavaScript 变量赋值。

1. =
2. +=

3. -=

案例:

```
let a = 30
let b = 20
console.log(a=100); // 结果: 100
```

```
console.log(a+=b); // a = a+b 结果: 50
```

```
console.log(a-=b); // a = a-b 结果: 10
```

7.3比较运算符

1. == 判断两边的值是否相等(但是不会验证数据类型是否一致) 如果相等返回true, 否则返回false 只要值相等就相等

```
<script>
let a = 30
let b = '30'
console.log(a==b); // 值相等就会相等 true
</script>
```

2. != 用来判断两个值是否不相等 如果不相等返回true, 否则返回false

```
<script>
let a = 30
let b = '30'
console.log(a!=b); // 值相等就会相等 false
</script>
```

3. === 判断两边的值 和数据类型是否相等 (值和数据类型都会进行验证)

```
<script>
let a = 30
let b = '30'
console.log(a===b); // 值相等就会相等 false
</script>
```

八、条件判断

2.1 if 判断

三个关键词汇：`if` `-- else` `-- else if`

语法结构：

```
if(条件){
    执行语句
}else if(条件){
    执行语句
}else{
    执行语句
}
```

```
<script>
let a = 30
if(a == 30){
    console.log('条件成立 执行下面语句');
}else{
    console.log('条件不成立 执行else的语句');
}
</script>
```

补充：

js中的 并且符号：`&&` 类似于python中的and

js中的 或者符号：`||` 类似于python中的or

案例：

```
//登录初始账号
let userName = "不渝" ;
let userPass = "1024" ;
//打印初始账户
console.log(userName+userPass)
// 采集用户信息
let loginName = prompt("请输入你的账户名称:")
let loginPass = prompt("请输入你的账户密码:")
// 登录验证
if(loginName==userName && loginPass==userPass){
    alert("登录成功！欢迎进入我的网页~")
}else{
    alert("登录失败！请刷新网页后重试！")
}
```

```
/*
题目：
判断是否为酒后驾车：
<20 正常驾驶
>= 20 < 80 饮酒驾驶
>= 80 醉酒驾驶
编写 js 程序判断是否为酒后驾车。
需求：1、能够从用户这里获取当前血液酒精浓度
```

```
2、能够有弹窗输出 当前 用户驾驶状态  
*/  
//从用户这里获取当前血液酒精浓度  
  
// 检查当前 用户驾驶状态 ,并弹窗输出
```

课后作业1:

完成课堂练习：查酒驾题目

课后作业2:

预习下节课的js内容（要求完成）

重点预习：引用数据类型、for循环、while循环、函数