

JavaScript

一、JavaScript的介绍

1、JavaScript的定义

JavaScript是运行在浏览器端的脚本语言，是由浏览器解析执行的，简称js。它能够让网页和用户有交互功能，增加良好的用户体验效果。

前端开发的三大块：1、HTML：负责网页结构。2、CSS：负责网页的样式。3、JavaScript：负责网页的行为，例如：网页与用户的交互效果。

二、JavaScript的使用方式

行内式，内嵌式，外链式

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width= , initial-scale=1.0">
  <title>Document</title>

  <!-- 内嵌式 -->
  <script>
    alert("我是内嵌式的js代码 ");
  </script>

  <!-- 外链式 -->
  <script src="js/main.js"></script>
</head>
<body>
  <!-- 行内式 -->
  <input type="button" value="按钮" onclick="alert('点我!')">
</body>
</html>
```

效果图：



三、JavaScript中的变量和数据类型

1、定义变量

JavaScript是一种弱类型语言，也就是说不需要指定变量的类型，JavaScript的变量类型由他的值来决定，定义变量需要用关键字"let"，一条JavaScript语句应该以";"结尾。

定义变量的语法格式:let 变量名 = 值;

```
let iNum = 123;
let str = 'asd';
```

2、数据类型

js中有六种数据类型，包括五种基本数据类型和一种复杂数据类型（object）。

5种基本数据类型：

- 1、number数字类型
- 2、string字符串类型
- 3、boolean布尔类型 true或false
- 4、undefined undefined类型，变量声明未初始化，它的值就是undefined
- 5、null null类型，表示空对象，如果定义的变量将来准备保存对象，可以将变量初始化null，在页面上获取不到对象，返回的值就是null

1种复合类型：

object：后面学习的JavaScript对象属于复合类型

3、输出方式：

alert('hello JavaScript!'); 在网页输出hello world

console.log('hello JavaScript!'); 在控制台输出hello JavaScript

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>

  <script>
    // 定义数字类型的变量
    let iNum1 = 1;
    let fNum2 = 2.5;
    // 定义字符串
    let sName = '张三';
    alert(iNum1);
    alert(fNum2);
    alert(sName);
    // 定义boolean类型
    let bIsOk = true;
    // 定义undefined类型
    let unData;
    // 定义空对象
```

```

    let oData = null;
    // 定义JavaScript对象类型
    let oPerson = {
        name: '张三',
        age: 20
    }
    // 查看数据类型使用typeof
    alert(typeof (iNum1));
    alert(typeof (fNum2));
    alert(typeof (sName));
    alert(typeof (bIsOk));
    alert(typeof (unData));
    // 由于js历史原因，设计之初没有null类型，null值归属于object类型，后续语言迭代后也没有进行修改
    // 对于大家，类型null值属于null类型
    alert(typeof (oData));
    alert(typeof (oPerson));
    // 我们还能获取哦Person对象中的值
    alert(oPerson.name);
    // 我们还能再控制台输出结果
    console.log(oPerson.name);
</script>
</head>
<body>
</body>
</html>

```

四、字符串拼接

1、字符串拼接

字符串拼接使用: "+"运算符

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <script>
        let sName = '李四';
        let iAge = 20;
        let sSex = '男';
        let sResult = sName + sSex;
        alert(sResult);
        // 字符串和数组进行拼接，底层自动把数字类型转换成字符串
        sResult = sName + iAge;
        alert(sResult);

        // +=
        iAge += sSex;
        alert(iAge);
    </script>
</head>
<body>

```

```
</body>
</html>
```

说明：

数字和字符串拼接会自动进行类型转换。

五、条件语句

1、条件语句的介绍

条件语句就是通过条件来控制程序的走向。

2、条件语句语法

- 1、if语句-只有当指定条件为true时，使用该语句来执行代码
- 2、if...else语句-当条件为true时执行代码，当条件为false时执行其他代码
- 3、if...else...if...else语句：使用该语句来判断多条件，执行条件成立的语句

3、比较运算符

假如x=5,查看比较后的结果：

比较运算符	描述	例子
==	等于	x == 8 为 false
===	全等（值和类型）	x === 5为true； x==="5"为false
!=	不等于	x != 8 为true
>	大于	x > 8 为false
<	小于	x < 8 为true
>=	大于或等于	x >= 8 为false
<=	小于或等于	x <= 8 为true

比较运算符示例代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>

  <script>
    let iNum = 5;
    let iStr = "5";
    // 在js里面如果数字和字符串进行比较，会自动把字符串转成数字类型然后进行比较
    if (iNum == iStr){
      alert("条件成立");
    }
    // 判断iNum和iStr的大小
```

```

        if (iNum < iStr){
            alert("条件成立");
        }else{
            alert("条件不成立");
        }

        // 定义一个成绩的变量
        let iScore = 80;
        // 使用if...else if ...else..结构进行判断成绩的等级
        if (iScore < 60){
            alert("不及格");
        }else if(iScore <= 70){
            alert("刚及格")
        }else if(iScore <= 80){
            alert("良好");
        }else{
            alert("优秀")
        }
    }
</script>
</head>
<body>

</body>
</html>

```

4、逻辑运算符

假如x=6, y=3, 查看比较后的结果:

逻辑运算符	描述	例子
&&	and	(x < 10 && y > 1) 为true
	or	(x == 5 y == 5)为 false
!	not	!(x == y)为true

逻辑运算符实例代码:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <script>
        let iNum1 = 1;
        let iNum2 = 2;
        // && 左右两边的条件都成立 结果才是true。if语句才会执行
        if(iNum1 >= 0 && iNum2 <= 2){
            alert("条件成立");
        }else{
            alert("条件不成立")
        }
    }

```

```

// || 左右两边的条件有一个成立 结果就是true, if语句才会执行
if(iNum1 >= 0 || iNum2 <= 2){
    alert("条件成立");
}else{
    alert("条件不成立")
}

// ! 表示对结果进行取反, true取反成false, false取反变成true
if(iNum1 >= 0 && iNum2 <= 2){
    alert("条件成立");
}else{
    alert("条件不成立")
}
</script>
</head>
<body>

</body>
</html>

```

六、循环语句

1、循环语句的介绍

循环语句就是让一部分代码重复执行, JavaScript中常用的循环语句有:

1、for。2、while。3.do-while。

2、for循环

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <script>
        let aArray = [1, 3, 9]
        // for循环数组
        // for(let index=0; index < aArray.length; index++){
        //     // 根据下标获取数据
        //     let ovalue = aArray[index];
        //     alert(ovalue);
        // }

        // while循环
        // 开始取值的下标
        // let index = 0;
        // while (index < aArray.length){
        //     // 根据下标获取数据
        //     let ovalue = aArray[index];
        //     alert(ovalue);
        //     index ++;
        // }
        let index = 0;
        // 首先执行一次循环体中的代码
    </script>

```

```
do {  
    let ovalue = aArray[index];  
    alert(ovalue);  
    index++;  
} while (index < aArray.length);  
  
</script>  
</head>  
<body>  
  
</body>  
</html>
```