

1. javascript概述

是一门基于对象的语言。

是一门独立的语言。

是一门脚本语言。脚本语言特点：直接解释执行。

是一门常用于浏览器端的语言，在后端也有技术的实现。Node.js

2. javascript

特点：

弱类型

脚本语言

基于对象

特性：

安全性

跨平台性

交互性

3. JavaScript不得不说的历史

和JAVA的关系：语法十分相似，但其实毫无关系。

1995年5月，Netscape，LiveScript

1995年12月，改名为JavaScript

1996年8月，微软，Jscript

1997-1999年，ECMA，ECMAScript，基于已有的JavaScript和Jscript，提出了标准的Script语法规则，JavaScript和Jscript都遵循这套标准。

4. JavaScript的引入

- a. 在head标签中书写一个script子标签，在引入的子标签内部书写javascript代码。

代码实现：

```
<HEAD>
  <title>js引入</title>
  <meta http-equiv="Content-type" content="text/html; charset=UTF-8"
  />
  <script type="text/javascript">
    alert("123");
  </script>
</HEAD>
```

- b. 在head标签中书写一个script子标签，在script起始标签身上使用src属性指

向一个js文件。虽然在script标签内部没有书写任何内容，但是**仍然不能够写成一个自闭标签**，如果变为自闭标签则会导致js代码失效。

代码实现：

```
<HEAD>
  <script type="text/javascript" src="1.js"></script>
</HEAD>
```

5. js语法特点

- 在javascript当中语句可以添加或不添加分号，都能够正常执行。
- 在浏览器中内置了一个js解析器，它会**自上而下的解析**js代码，并且代码如果未添加分号，会**自动添加分号保证语句正常执行**。
- 建议书写上分号，避免java和js语言切换使用时出现不必要的错误。
- 定义变量时使用var，弱类型。

注意：可以利用console.log()方法在浏览器F12的控制台中打印数据。

6. js语法-数据类型

基本数据类型和复杂数据类型。

a. 基本数据类型：五种

Number数值型 String字符串型 boolean 布尔型 Undefined未定义类型 Null 空类型

- Number类型是js的一个基本数据类型，还是js中的一个包装对象。

属性和方法：

- Number.MAX_VALUE
- Number.MIN_VALUE
- Infinity
- Infinity
- NaN（非数字）

注意：NaN属性和任何一个类型的对象都不相等，包括它本身。

toString()

valueOf()

- String类型是js的一个基本数据类型，还是js中的一个包装对象。

属性和方法：

详见api。

- boolean类型是js的一个基本数据类型，当前类型中只有两个值，true和false。还是js中的一个包装对象。

||

&&

|

&

!

注意：| &会在js中作为计算使用。true会置换为1，false会置换为

0.

- iv. Undefined类型是js的一个基本数据类型。当前类型中只有undefined这一个值。表示未定义的变量。
- v. Null类型是js的一个基本数据类型。当前类型中只有null一个值。表示当前值不存在。

拓展：null和undefined

`console.log(null == undefined);`//true js中认为null和Undefined出于统一类型，所以相等。

`console.log(null === undefined);`// false ===绝对相等，判断两个对象身上的值是否相等。

b. js语法--数据类型自动转换

js在需要时会自动对类型进行转换，转换的规则：

数字	可以在需要时转换为对应的字符串形式。0会转换位false，其他数字会转换为true.在需要对象时可以自动转换位Number对象。
字符串	可以在需要时转换位对应的数值，需要注意当数值和字符串进行加法运算时，会处理为字符串的拼接，所以需要通过parseInt或parseFloat将字符串强制转换为数值类型参与运算。非空字符串转换为ture，空字符串转换为false。需要对象时自动转换位String对象
布尔类型	true转换位1，false转换位0.转换为字符串的"true"和"false".转换位Boolean对象
对象类型	如果为null则转成字符串"null",如果为null转换为false

c. js语法--运算符

+	加法运算。拼接运算。
-	减法中如果有字符串类型的数据，会尝试转换成数值型进行运算。
/	在运算中如果出现小数则仍然保留。

注意：在运算过程中，如果结果为小数则保留小数。如果运算参数有小数结果为整数，则结果要保留整数。

i. typeof()运算符：

```
var a = 0;
var b = "abc";
var c = true;
var d = new Object();
console.log(typeof(a) == "number");//number
console.log(typeof(b));//string
console.log(typeof(c));//boolean
console.log(typeof(d));//object
```

三目运算符:

`2+3>5?"true":"false"`

d. 定义变量

- i. 在js中使用var定义变量，定义的变量类型为弱类型。

弱类型指的是当前变量没有类型，可以被任意类型的数值所赋予。

```
var a = 0;
a = "abc";
a = true;
a = new Object();
console.log(a);
```

- ii. 变量的作用范围 局部变量和全局变量。

```
function mx() {
    i = 0;
}
```

`mx();`

`console.log(a);`

定义变量的时候使用var，则当前变量为一个局部变量，只在当前代码块范围内有效。

定义变量的时候不使用var，则当前变量为一个全局变量，整个script范围内都可以使用。

7. js语法-语句

- a. for循环 没有增强for循环

- b. while

- c. do while

- d. switch case

-----和java的使用方式一样

- e. if语句

```
var x = 4;
if(x = 5){
    console.log("x = 5");
}else{
    console.log("x != 5");
}
```

上述语句判断中没有书写两个等号，形成了一个赋值的语句，导致语句的执行变为先赋值再判断，影响了结果。应该避免这种情况。

注意：将常量放在等号之前即可避免这个问题。

8. js语法-复杂数据类型：函数、数组、对象

- a. js函数是一段js代码的合集。调用函数时需要书写函数名称和一对小括号，完成调用。书写的这个合集本质是一段字符串。

- b. js函数在调用的时候，可以在传入比函数原有参数数量更多或更少的参数。如果传入的参数数量少于参数列表，则未被传入的参数定义为undefined。如果传入的参数数量多于参数列表，则多的参数可以使用arguments属性来遍历获取。
- c. 函数在执行的时候需要在函数名后添加一对小括号。函数也可以当前变量使用，这是指数写函数名即可。
- d. js函数是一个内置对象Function。

e. 函数的定义

i. 普通函数定义方式

```
function mx(参数列表) {  
    函数体  
}
```

mx()

示例：

```
function mx(a,b){  
    return a + b;  
}  
console.log(mx(1,2));
```

ii. 动态函数定义

```
var mq = new Function("a","b","return a+b");  
console.log(mq(2,3));
```

iii. 直接量定义函数(匿名函数)

js中函数可以看成是一个字符串，所以函数形式如何，就直接书写，利用这种形式定义字符串就是一个函数。

```
var ma = function(a,b){  
    return a+b;  
}  
console.log(ma(4,5));
```

9. js语法-数组

在js中数组是由一对中括号组成，其中可以包含多个不同类型的元素，每一个元素之间使用逗号隔开，这种形式的数据就是称之为数组。数组本质上是一段字符串。

a. 定义数组

i. var arr = new Array();

ii. var arr = new Array(3);

iii. var arr = new Array(1, "b", true, new Object());

iv. 直接量定义数组

```
var arr = [1,2,3];
```

b. 操作数组

i. 向指定下标添加元素

`arr[index]`

ii. 移除元素

`pop()` 最后一个

`shift()` 第一个

iii. 添加元素

`push()`

iv. 遍历元素

for循环

注意：js中数组的长度是任意的。与初始时定义的长度无关。

注意：js中数组中存储的数据类型是任意的。

10. js语法-对象

a. js对象分为两大类**内置对象**和**自定义对象**

o **js的内置对象：**

String -- 基本数据类型 字符串类型 的包装对象

Boolean -- 基本数据类型 布尔类型 的包装对象

Nubmer -- 基本数据类型 数值类型 的包装对象

Array -- 数组类型 的包装对象

Function -- 函数类型 的包装对象

Math -- 数据对象，封装了很多数学常量和数学方法

Date -- 日期时间对象，封装了很多和日期实现相关的方法

Global -- 全局对象。js中有一些方法和属性经常使用，但归到哪个对象上都不合适，所以js中有一个Global对象整合了这些方法和属性。Global中定义的方法和属性特点是属于全局，可以直接使用。

思考：`parseInt("123abc123");`打印结果。结果为 123

★1) `isNaN()`可以判断当前出入的值是否为一个非数字。如果返回true。则表明当前值是一个非数字。如果返回false，则表明当前值不是一个非数字(是数值类型)

RegExp --正则对象，保存有关正则表达式模式匹配信息的固有全局对象。Partten 邮箱正则。

o **自定义对象**

在js中对象由一对大括号组成，其中包含一个或多个元素。每个元素都是由键值对的形式组成，键名和值之间使用冒号隔开。两个元素之

间使用逗号隔开，这种形式的数据就是称之为对象。对象的本质是字符串。

i. 无参构造函数定义对象

```
function Person(){  
  
}  
var p = new Person();
```

ii. 有参构造函数定义对象

```
function Person(name,age){  
    this.name = name;  
    this.age = age;  
}
```

iii. 直接量定义对象。

```
var p = {name:"lishuai",age:18}
```

b. 操作对象的api:

```
//添加属性/方法  
p.height = "180cm";  
//获取属性  
console.log(p.name);  
console.log(p["name"]);  
//删除属性  
delete p.name;  
console.log(p);  
//with语句  
with(p){  
    console.log("with获取:"+age);//p.age  
}
```

▪ for...in语句:用来遍历对象的所有属性的名称

```
var p = {name:"liubei",age:19,sleep:function()  
{alert(this.name+"sleep....")}};  
for(var x in p){  
    alert(x);  
}
```

11. JSON

JSON本质上就是一段字符串，能够保存较复杂关系的数据，具有良好的数据保存格式，又极为轻量，加之多种代码平台都支持对字符串的处理，所以我们可以使用JSON字符串进行数据的传入，甚至跨平台传输。

```
var data ={
    name:"曹洋",
    age:18,
    addr:"安徽",
    girlfriends:[
        {name:"貂蝉",job:"跳舞",age:18},
        {name:"西施",job:"唱歌",age:19}
    ]
}
console.log(data["girlfriends"][1]["job"]);
```

拓展：

JSON对象：

```
var data ={
    name:"曹洋",
    age:18,
    addr:"安徽",
    girlfriends:[
        {name:"貂蝉",job:"跳舞",age:18},
        {name:"西施",job:"唱歌",age:19}
    ]
}
```

JSON字符串：

```
var data ={
    "name":"曹洋",
    "age":"18",
    "addr":"安徽",
    "girlfriends":[
        {"name":"貂蝉","job":"跳舞","age":"18"},
        {"name":"西施","job":"唱歌","age":"19"}
    ]
}
```

两者的区别：json对象的属性名称没有双引号。json串属性名称和属性值都必须使用双引号。

两者之间可以转换，使用Sun公司体用的JSONUtils工具类即可。

1. DHTML概述

- a. DHTML Dynamic动态的超文本标记语言。
- b. 在HTML中内置了一些js相关的接口，允许js把HTML页面中的标签当做对象处理。
- c. DHTML将整个HTML元素当做js对象来处理，形成了一颗js的对象树。
- d. 操作js的对象树就是在操作HTML页面中的元素。

2. BOM和DOM

BOM browser object model 浏览器对象模型

- window:代表一个浏览器窗口的对象

- 其中包含的方法：

onblur:失去焦点

onfocus:获得焦点

!!!onload:当前浏览器页面装载完成后触发

!!!alert

!!!confirm

!!!prompt

!!!close (//仅限ie浏览器，chrom和firefox需要解决浏览器禁止js关闭非js创建页面的问题)

!!!setInterval

!!!setTimeout

setInterval 和 setTimeout的不同？

- 其中包含的对象：(我们可以通过window引出这些对象)

location

screen

history

navigator

document

- location

!!!href 获取或设置地址栏上的地址。通过此属性js可以控制浏览器访问一个新的地址。

- navigator

- history

length

back()

forward()

go()

DOM document object model 文档对象模型