JS变量/数据类型

2020年4月5日 18:50

变量：

- Var 变量名；

var 变量名；
赋值：
变量名=值;

```
var  userName;
var  bookPrice = 25.5;
```

声明时不要省略var关键字，否则声明的是全局变量

1、声明一个变量，保存 学员的年龄,值为 25

```
var age;
age=25;
```

- 一次性声明多个变量名：

```
var 变量名1,变量名2,...变量名n;
var stuName= "PP.XZ",stuAge=25, stuHeight;
        等同于
        var stuName= "PP.XZ";
        var stuAge=25;
        var stuHeight;
```

```
var name1, name2, name3;
var age1, age2=30;
```

- 变量名命名规范：

1、不允许使用JS的关键字和保留关键字
2、由字母、数字、下划线以及$组成

关键字

break	case	catch	continue	default
delete	do	else	false	finally
for	function	if	in	instanceof
new	null	return	switch	this
throw	true	try	typeof	var
void	while	with	undefined	...

```
        var stuName;
        var stuName1;
        var stu_name;
        var stu$name;
```

3、不能以数字开头

```
        var stuName1;
        var 1stuName;错误
        var $stuName;正确
```

4、尽量见名知意

```
        var a,b,c,d,ab,ac,ad;
        var stuName,stuAge; 更推荐
```

5、可以采用"驼峰命名法
变量名为合成词时，第一个单词全小写，从第二个单词开始，每个单词首字符变大写

```
        var stuName;
```

如果只有一个单词作为变量名，全小写

```
        var age;
```

- 变量的使用：

直接输出小猪佩奇

```
var stuName="小猪佩奇";
console.log(stuName);
var stuAge;
console.log(stuAge);
```

变量声明后，从未赋值，称之为未经初始化变量

- 对变量进行存取操作：

1、获取变量的值-GET操作

```
var userPwd = '123456' ;

console.log( userPwd );
document.write( userPwd );
var newPwd = userPwd;
```

2、保存(设置)变量的值 - SET操作

```
var oldPwd = '123' ;

oldPwd =  '456' ;
oldPwd = newPwd;
```

运算符和表达式：

- 算术运算符：

2.1、算术运算符

加(+)、 减(-)、 乘(*)、除(/)、求余(%)

```
- 可以表示减号，也可以表示负号，如：x=-y
+ 可以表示加法，也可以用于字符串的连接
```

ex:

```
var num1 = 15;
var num2 = 18;
var str1 = "15";

console.log(num1+num2);//33
console.log(num1+str1);//1515
```

% : 取余操作，俗称 模
作用：取两个数字的余数
ex:

```
var  i = 10 % 3;                    //i值为1
```

- 自增和自减：

++ : 自增，在数值的基础上，进行+1操作
-- : 自减，在数值的基础上，进行-1操作

```
i ++ ;                    //相当于 i = i + 1;

i -- ;                    //相当于 i = i - 1;

i = 1;
j = i++ ;                 // i结果为2，j结果为1

i = 1 ;
j = ++ i ;                // i结果为2，j结果为2
```

- 关系运算符：【全等/不全等】

```
> : 大于
< : 小于
>= : 大于等于
<= : 小于等于
== : 判断等于
        注意：不比较类型，只比较数值
!= : 不等于
=== : 全等
        注意：除数值之外，连同类型也会一起比较
!== : 不全等
```

关系表达式的运算结果为 boolean类型(true 或 false)

```
var input=prompt("请输入一个数据：");
判断 input 是否为 数字??

isNaN(数据)
isNaN()会抛开数据类型来判断数据是否为数字
如果 数据 是数字类型，则返回 false
如果 数据 不是数字类型，则返回 true
```

- 逻辑运算符：

1. 逻辑与：&&

1）逻辑与(&&)

关联两个条件，两个条件都为真的时候，那么整个表达式的结果才为真语法： **条件1 && 条件2**

2. 逻辑或：||

2）逻辑或(||)

关联两个条件，两个条件中，只要有一个为真，那么整个表达式的结果就为真

3. 逻辑非：！

```
对条件取反
注意：逻辑非，只有一个操作数
语法：!条件

非真即假
非假即真
```

- 条件运算符：

1. 三目运算符：

```
表达式1?表达式2:表达式3;
    表达式1是一个条件，值为boolean类型
    若表达式1的值为 true，则执行表达式2的操作作为整个表达式的结果
    若表达式1的值为false，则执行表达式3的操作作为整个表达式的结果
```

```
var score = 85;
var result = score >= 80 ?  "优秀" :  (
                score>=60 ? "合格" : "不合格"
);
```

数据类型：

1. 原始类型（基本类型）
   - 数字类型

```
number类型
    可以表示32位的整数以及64位的浮点数
    整数：32位即4字节
    浮点数：即小数，64位，8字节
        整数：
            十进制：生活中常用数字
            八进制：逢八进一
                var n1=0123;
            十六进制：逢16进1
                0-9 A-F 组成
                var n2 = 0x123;
            小数：
                var n1 = 34.56;
                var n2 = 4.5e10;
```

   - 布尔类型

```
作用：
    用于表示 条件的结果
取值：
    true：真，肯定的结果
    false：假，否定的结果
        ex
            var r1 = true;
            var r2 = false;
```

数据类型的转换：

- 隐式转换：自动转换，js自己进行转换操作

1）函数

```
typeof() 或 typeof
ex:
        var num1 = 15;
        var s = typeof(num1); : 获取 num1 的数据类型
        var s1 = typeof num1; : 获取 num1 的数据类型
```

2）NaN

```
Not a Number
不是一个数字
isNaN(数据) : 判断 数据是否为 非数字
是不是一个数字
结果为 boolean 类型
结果为 true ：不是一个数字
结果为 false：是一个数字
```

注：所有的数据类型 与 string 做 + 运算时，最后的结果都为 string

- 强制转换：

1、转换成字符串类型：

toString() 将任意类型的数据转换为 string 类型变量.toString();

true：真，肯定的结果
false：假，否定的结果

ex

```
var r1 = true;
var r2 = false;
```

- 字符串类型

  转义字符：

  1、\n　换行

  2、\r　回车

  3、\t　一个制表符

- 未定义类型

  undefined

  1、声明变量未赋值

  2、访问对象不存在的属性

- 空类型

  NULL

2. 引用类型

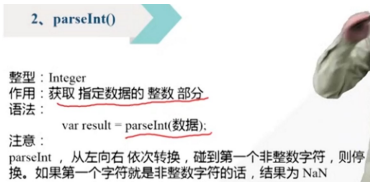
- 强制转换：

1、转换成字符串类型：

toString()　将任意类型的数据转换为 string 类型变量.toString();

会得到一个全新的结果，类型为 string

2、获取数据整数部分：



3、转换成小数，获取小数部分：

parseFloat()　Float：浮点类型->小数

作用：将 指定数据转换成 小数

语法：var result = parseFloat(数据);

```
var result=parseFloat("35.25"); //35.25
var result=parseFloat("35.2你好!");//35.2
var result=parseFloat("你好35.2");//NaN
```

4、数字字符解析为数字

Number()　作用：将一个字符串解析为number

语法：var result=Number(数据);

注意：如果包含非法字符，则返回NaN