

## 第六课 数据类型

### 学习目录

- **typeof 操作符**
- **Undefined 类型**
- **Boolean 类型**
- **Number 类型**
- **String 类型**
- **Object 类型**
- **Null 类型**

### ①typeof 操作符

typeof 是一元运算符，操作符返回一个字符串，表示未经计算的操作数的类型。

数据类型	字符串	描述
Undefined 类型	undefined	未定义或未声明的变量
Boolean 类型	boolean	布尔值真或假
String 类型	string	字符串
Number 类型	number	数值
Object 类型/Null 类型/Array	object	对象/null/数组
Object 类型	function	函数

### ②Undefined 类型

Undefined 类型只有一个值，即特殊的 undefined 值。在使用 var 声明变量但未对

其加以初始化时。

```
var wrap;
```

```
alert(wrap == undefined); //true
```

未定义初始化的变量与未声明的变量不一样。

```
var wrap;
```

```
alert(wrap); //报错
```

**Ps：**不过用 `Typeof` 操作符检测未定义和未声明变量的数据类型返回的字符串都是 `undefined`，因此在声明一个变量之后尽量要给这个变量赋值，避免类型检测 `typeof` 出现问题。

```
var abc;
```

```
alert(typeof abc); //undefined
```

```
alert(typeof ccc); //undefined
```

### ③ Boolean 类型

布尔 Boolean 类型表示逻辑实体，它只有两个值，保留字 `true` 和 `false`，分别代表真和假这两个状态。

在实际运用中，一般不直接使用这两个值，而是通过一系列的运算和转换把其他的值当做布尔值来使用，布尔类型主要应用于以下代码中：

1. 通过条件和循环语句得出布尔值进行逻辑判断
2. 通过关系运算符用于判断两个值之间的关系，根据关系是否存在而返回 `true` 或 `false`

**ps：**通过 `Boolean()` 可以把其他数据类型强制转换成布尔类型。

## 类型转换（布尔值）

数据类型	转换为 true 的值	转换为 false 的值
Undefined 类型	不适用	undefined
Boolean 类型	true	false
String 类型	任何非空字符串	空字符串
Number 类型	任何非零数字值（包括无穷大）	0 和 NaN
Object 类型	任何对象	null

## ④Number 类型

### 1.进制整数值

```
var num = 123;//十进制整数 123
```

```
var num = 0b100000; //二进制整数 32
```

```
var num = 061;//八进制整数 49
```

```
var num = 0F1;//十六进制整数 241
```

ps：进行相关算术计算八进制与十六进制的数值会先转换成十进制并进行计算。

### 2.浮点数值

浮点数就是该数值中必须包含一个小数点，并且小数点后面必须有一位数字。浮点数的最高精度是 17 位小数，浮点数虽然精度高，但是计算精确度却远不如整数。

```
var float1 = 1.11;
```

```
var float2 = 0.33;
```

极大或极小的数字可通过科学计数法来写：

```
var y=123e5;    // 12300000
```

```
var z=123e-5;   // 0.00123
```

### 浮点数的精度问题

浮点数计算会产生舍入误差的问题， $0.1+0.1$  不一定等于  $0.3$ 。这是因为 js 中的浮点数的计算是基于 IEEE754 二进制浮点数算术标准来做的，其他的语言也有类似问题。

```
var float1 = 0.1;
```

```
var float2 = 0.2;
```

```
alert(float1 + float2);//0.30000000000000004
```

### 3.数值范围

因为内存限制，所以 js 中并不能保存所有的数值。

Number.MAX\_VALUE 最大值  $1.7976931348623157e+308$

Number.MIN\_VALUE 最小值  $5e-324$

如果一个数值是正数并且超出最大值就会用正无穷 Infinity 表示，如果一个数值是负数也超出最小值就用负无穷-Infinity 表示。

**isFinite()判断正负无穷之间的数值为 true。**

Number.POSITIVE\_INFINITY 保存了正无穷。

Number.NEGATIVE\_INFINITY 保存了负无穷。

### 4.NaN

NaN 是一个非数值的特殊数值。这个数值用来表示一个本来要返回数值的操作数未返回数值的情况，避免程序报错。

```
var num = 0 / 0;//NaN
```

任何与 NaN 有关的操作都会得到一个 NaN，NaN 不等于任何值，也不等于他自己。

isNaN()用来判断一个值是不是 NaN。

## 5.数值转换

其他数据类型转换成数值 Number 类型。

### Number()

Number() 函数把需要转换的值转换为数字，如果对象的值无法转换为数字，那么

Number() 函数返回 NaN。

**Boolean 类型 true 转行成 1，false 转换成 0**

**Number 类型 直接转换成数字**

**Null 类型 转换成 0**

**Undefined 类型 转换成 NaN**

**Object 类型 转换成 NaN**

**String 类型**如果字符串为空，转为 0，如果字符串中是纯数字，数值会直接转为十进制的数值，其他字符串返回 NaN。

### parseInt()

parseInt() 函数可转黄一个字符串，并返回一个整数。

**只有字符串中的第一个数字会被返回**

**开头和结尾的空格是允许的**

如果字符串的第一个字符不能被转换为数字，那么 `parseInt()` 会返回 `NaN`。

```
parseInt("40 years");//返回 40
```

```
parseInt("He was 40");//返回 NaN
```

这个方法第二个参数表示要转换的数字的进制基数。

```
parseInt("123");          //返回 123
```

```
parseInt("23",10);        //返回 23
```

```
parseInt("11",2);         //返回 3 (2+1)
```

```
parseInt("16",8);         //返回 14 (8+6)
```

```
parseInt("1f",16);        //返回 31 (16+15)
```

### `parseFloat()`

`parseFloat()` 函数可转换一个字符串，并返回一个浮点数。

如果字符串的第一个字符不能被转换为数字，那么 `parseFloat()` 会返回 `NaN`。

`parseFloat()` 会忽略前导的 0

`parseFloat()` 只能转换十进制值

```
parseFloat("6.66.31");//返回 6.66
```

```
parseFloat("006.66.31");//返回 6.66
```

```
parseFloat("666orange");//返回 666
```

## ⑤String 类型

字符串是用来表示零或多个字符组成的字符序列，也就是字符串。在 js 中的字符串需要使用引号引起来(英文模式下单引号或者双引号都可以)。不可以一边单引号，另一边双

引号，会报错。

```
var str = '星星课堂' ;//报错
```

字符串中会有一些特殊字符，这些字符不一定会被打印显示出来，但是可能会有其他作用，比如换行、退格等等，表示这些特殊字符需要通过转义。

代码	输出
\'	单引号
\"	双引号
\&	和号
\\	反斜杠
\n	换行符
\r	回车符
\t	制表符
\b	退格符
\f	换页符

字符串 length 属性可以返回字符串的长度。

```
var str = '星星课堂真好' ;
```

```
alert(str.length);
```

### 字符串的简单拼接

字符串可以通过加号操作符来进行拼接，其他数据类型也可以用加号运算符与一个字符串拼接转换成字符串。

```
alert(123+ " ");
```

### 字符串转换

其他数据类型转换成数值 String 类型。

#### toString()

任何数值、布尔值、对象和字符串本身都有这个 toString 方法，null 与 undefined 没有，

数值类型在转换时还可以传入一个进制基数为这个方法的参数。

```
var num = 123;
```

```
alert(num.toString());
```

#### String()

在不确定要转换的变量值的数据类型的情况下，通过 String()也可以把这个值转换成字符串，

null 转换成 'null'，undefined 转换成 'undefined'。

```
var obj = null;
```

```
alert(String(obj));
```

## ⑤Object 类型

对象是一组数据和功能的集合，用来存放一些需要使用的数据信息和功能方法。比如说在实际生活中，一部手机是一个对象，因为他包含各种属性参数和功能操作，js 中的对象也是这个道理。对象包含很多值，对象的写法是以名称:值对的方式来书写（名称和值由冒号分隔）。

### 对象的定义

使用 new 操作符来定义一个对象 var obj = new Object();



使用对象字面量来定义一个对象 `var obj = {};`

### 对象的属性

```
var phone = {name:"huawei", model:"meta20", price:200};
```

### 对象的方法

```
var phone = {  
  
    playGame:function(){...},  
  
    callNumber:function(){},  
  
    sendMessage:function(){...}  
  
};
```

## ⑤Null 类型

null 就是完全空的对象，typeof 操作符检测 null 值时会返回"object"。如果定义的变量准备在将来用于保存对象，应该将该变量初始化为 null。

**ps：undefined 其实是来自 null 的一种形式，因此它们的相等表达式返回 true。**

## 谢谢观看！

如果你自己难以解决的问题，请发送邮件到 [xingxingclassroom@aliyun.com](mailto:xingxingclassroom@aliyun.com)，并备注

**星星课堂问题咨询**这几个字+问题介绍为邮件标题，谢谢。

我是星星课堂老师：周小周