

数据类型

2017年6月27日 16:14

1 js数据类型：

两大类型：

基本（简单）数据类型

复合（引用）数据类型

基本（简单）数据类型

- **String** => 字符串 值：一对单引号或双引号包含的0个或多个字符串组成串
- **Number** => 数字 整数 浮点数
取值范围：
正无穷大 +Infinity
负无穷大 -Infinity
- **Boolean** => 布尔值 值： true false
- **Undefined** => 未定义
值：undefined
1. 定义一个变量没赋值，默认的值undefined
2. 找一个对象上不存在的属性，返回undefined
- **Null** => 空
值：null
声明变量不确定要赋的值，可以使用null来占位

复合（引用）数据类型

- 动态特性：

getElementsByTagName

getElementsByTagName 有动态特性

当向一个元素增加或删除元素，集合也会改变

querySelector

querySelectorAll

没有动态特性

使用这两个方法的时候，要等动态添加元素之后再获取

```
// js中表示的最大的数
```

```
console.log( Number.MAX_VALUE );
```

```
// js中表示的最小的数
```

```
console.log( Number.MIN_VALUE );
```

Object => 对象

值:是由0个或多个名/值组成的集合

内置的函数

字符串 String

数字 Number

内置的函数, 会打印function 函数名() { [native code] }

布尔值 Boolean

2 检测数据的类型

一元运算符 typeof

使用: typeof 数据

typeof运算之后会返回一个字符串, 这个字符串来表示数据的属于的类型

typeof 数据 运算数据之后, 可以得到以下字符串:

string
number
boolean
undefined
object
function

数组和函数都属于对象的

函数是可执行的一种对象, 比较特殊

js中函数属于一等公民

注意: 使用typeof检测null, 返回的是object这个字符串

+ 二元运算符

三元运算符 条件 ? 条件成立走这里 : 条件不成立走这里

3 数据类型转换

把一种数据的类型转成另一种类型

把数据转成三种类型:

Number

- 显式类型转换

开发者自己手动调用方法进行转化

把数据转成三种类型：

Number
String
Boolean

- 转成数字类型：

Number(要转的数据)

true => 1

false => 0

纯数字的字符串 => 转对应的数字

undefined => NaN

null => 0

NaN => Not a Number 数字类型

调用Number之后，不能转对应的数字，会用NaN来表示

运算失败后，会用NaN表示

和任何值比较都不相同，包括自己比较也不相同

和任何值比较都返回false

检测一个值是不是NaN

可以使用NaN自己和自己不相同这个规则来比较

```
if( n != n){
```

```
    console.log( "我是NaN" );
```

```
}
```

isNaN(值)

值是NaN，返回true

- 显式类型转换

开发者自己手动调用方法进行转化

String Number parseInt Boolean

- 隐式类型转换

解析器为了运算方便，会隐式把一种类型转成另一种类型
+

某一种数据类型会提供不同的方法，这些方法
方便操作这类数据

字符串.charAt(下标)

查找字符串中指定下标的字符

```
var str = 'abcd';  
// 下标 从0开始  
console.log( str[2] );  
// 使用方法找  
console.log(str.charAt(1));
```

值不是NaN, 返回false

备注:

函数 => 方法

`parseInt`(要转的数据) 得到的是整数

要转的数据遇到不是数字的就返回数字部分

数据的第一位不是数字, 返回NaN

`parseFloat`(要转的数据) 得到的是小数

- 转成字符串类型

`String`(要转的数据)

`''`+数据

加号+:

数字运算

字符串拼接

只要有一边是字符串类型, 把另一边转成字符串类型的

当转成字符串类型的时候, 内部会调用一个方法, `toString()`

对象是通过一个函数创建出来的。

数字 => 字符串

```
console.log( String(123) ); // "123"
```

布尔值转字符串

```
console.log( String(true) ); // 'true'
console.log( String(false) ); // 'false'
```

对象转字符串 会返回一个字符串[类型, 构造函数]

```
console.log( String({}) ); // [object Object]
```

```
null      console.log( String(null) ); // 'null'
undefined console.log( String(undefined) ); //'undefined'
```

- 转成布尔值

`Boolean(要转的数据)`

`!` 取反

先把数据转成布尔值，取反

`!!` 转成布尔值

那些能转成false

`'' 0 undefined null false`