

数据类型分类

2020年02月16日, 星期日 17:39

1.ES5 数据类型 (6种) 及其划分 (2类)

基本 (原始) 类型 (Number、String、Boolean、Null、Undefined)

引用 (对象) 类型 (Object (Array、Function、Date等))



Object对象:

思考: 如何判断数据的数据类型? Typeof

2.typeof——判断数据类型

● typeof 操作符

— 返回一个字符串, 表示未经计算的操作数的类型。

— **typeof operand** or **typeof (operand)** ← 括号可选

类型	结果
Undefined	"undefined"
Null	"object"
Boolean	"boolean"
Number	"number"
String	"string"
函数对象	"function"
任何其他对象	"object"

如何区分 Array、Date 等对象呢?

使用typeof无法检测出Null和任何其他对象这些都属于object类型的对象。

未经计算的操作数类型:

```
> var a = 1; //number
var b = "hello"; //string
< undefined
> a + b;
< "1hello"
> typeof a;
< "number"
>
```

3.数据、变量和类型

JS中的变量没有类型

JS中的数据值有类型

变量可以随时持有任何类型的值

在对变量执行 `typeof` 操作时，得到的结果并不是该变量的类型，而是该变量持有的值的类型。

```
typeof "42"; //string
typeof 42; //number
var a = "42"; typeof a; //string
var a = 42; typeof a; //number
```

4.练习：写出遍历 `person` 对象中的所有属性的代码，并在控制台输出除方法外所有的属性的属性值。

```
var person = {
  name: "Lily",
  age: 20,
  singing: function() {
    console.log('singing');
  },
  playing: function() {
    console.log('playing');
  }
};
```

答：

```
for (var a in person) {
  console.log(person[a]);
}
```

//`person`是一个对象 你代码中的`a`是属性名字的代指 访问属性值的方式是 对象[属性名]，所以 `person[a]`输出的是真正的属性值

5.函数没有`return`值的时候默认返回值类型为`Undefined`。

```
function Student(name, age) {
  this.name = name;
  this.age = age;
}
var stu = Student("Lucy", 20);
console.log(typeof stu);
```

请问在控制台输出结果为（ ）。

- ☐ A string ☐ B undefined
- ☐ C null ☐ D object