**数组**是一组按顺序排列的集合，集合的每个值称为元素。JavaScript的数组可以包括任意数据类型。例如：

[1, 2, 3.14, 'Hello', null, true];

**对象**

JavaScript的对象是一组由键-值组成的无序集合，例如：

**var** person = {

name: 'Bob',

age: 20,

tags: ['js', 'web', 'mobile'],

city: 'Beijing',

hasCar: true,

zipcode: null

};

JavaScript对象的键都是字符串类型，值可以是任意数据类型。上述person对象一共定义了6个键值对，其中每个键又称为对象的属性，例如，person的name属性为'Bob'，zipcode属性为null。

要获取一个对象的属性，我们用对象变量.属性名的方式：

person.name; // 'Bob'

person.zipcode; // null

### 变量

变量的概念基本上和初中代数的方程变量是一致的，只是在计算机程序中，变量不仅可以是数字，还可以是任意数据类型。

在JavaScript中，使用等号=对变量进行赋值。可以把任意数据类型赋值给变量，同一个变量可以反复赋值，而且可以是不同类型的变量，但是要注意只能用var申明一次，例如：

**var** a = 123; *// a的值是整数123*

a = 'ABC'; *// a变为字符串*

JavaScript在设计之初，为了方便初学者学习，并不强制要求用var申明变量。这个设计错误带来了严重的后果：如果一个变量没有通过var申明就被使用，那么该变量就自动被申明为全局变量：

i = 10; // i现在是全局变量

不用var申明的变量会被视为全局变量，为了避免这一缺陷，所有的JavaScript代码都应该使用strict模式。我们在后面编写的JavaScript代码将全部采用strict模式。

**由于JavaScript的对象是动态类型，你可以自由地给一个对象添加或删除属性：**

**var** xiaoming = {

name: '小明'

};

xiaoming.age; *// undefined*

xiaoming.age = 18; *// 新增一个age属性*

xiaoming.age; *// 18*

**delete** xiaoming.age; *// 删除age属性*

xiaoming.age; *// undefined*

**delete** xiaoming['name']; *// 删除name属性*

xiaoming.name; *// undefined*

**delete** xiaoming.school; *// 删除一个不存在的school属性也不会报错*

如果我们要检测xiaoming是否拥有某一属性，可以用in操作符：

var xiaoming = {

name: '小明',

birth: 1990,

school: 'No.1 Middle School',

height: 1.70,

weight: 65,

score: null

};

'name' **in** xiaoming; // **true**

'grade' **in** xiaoming; // **false**

不过要小心，如果in判断一个属性存在，**这个属性不一定是xiaoming的，它可能是xiaoming继承得到的：**

'toString' **in** xiaoming; // **true**

**要判断一个属性是否是xiaoming自身拥有的，而不是继承得到的，可以用hasOwnProperty()方法**：

var xiaoming = {

name: '小明'

};

xiaoming.hasOwnProperty('name'); // **true**

xiaoming.hasOwnProperty('toString'); // **false**

### 多行条件判断

如果还要更细致地判断条件，可以使用多个if...else...的组合：

**var** age = 3;

**if** (age >= 18) {

alert('adult');

} **else** **if** (age >= 6) {

alert('teenager');

} **else** {

alert('kid');

}