1. JS中7种内置类型：

空值（null）

未定义（undefined）

布尔值（Boolean）

数字（number）

字符串（string）

对象（object）

符号（symbol，ES6中新增）

除对象外，其他统称为“基本类型”

1. （1）使用typeof运算符来查看值的类型，他返回的是类型的字符串值

Typeof null==”object”;除此之外，其他6种类型均有同名的字符串值与之对应

（2）typeof function a(){…….}===”function”;//true

A、函数是“object”的一个子类型，函数是“可调用对象”，他有一个内部属性[[call]]，该属性使其可以被调用

B、函数不仅是对象，还可以用有属性

函数对象的length属性是其声明的参数个数

(3)数组也是对象，也是“object”的一个子类型；数组中的元素按数字顺序来进行索引（而非普通像对像那样通过字符串键值），其length属性是其元素个数

1. JS种变量没有类型，只有值才有，变量可以随时此有任何类型的值

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

1. Undefined和undeclaerd

（1）已在作用域中声明但没有赋值的变量，是undefined；相反，还没有在作用域中声明过的变量是undeclared

（2）typeof undeclared变量，会返回“undefined”

（3）JS将二者混为一谈，在我们试图访问“undeclared”变量时这样报错：ReferenceError:a is not define.通过typeof的安全防范机制（阻止报错）来检查undeclared变量，又是是个不错的办法

1. 数组
2. 在JS中，数组可以容纳任何类型的值，可以是字符串、数字、对象，甚至是其他数组（多维数组是同故宫这种方式实现的）
3. 对数组声明后即可向其中加入值，不需要预先设定大小
4. 使用delete运算符可以将单元从数组中删除，但是请注意，单元删除后，数组的length属性并不会发生变化
5. 在创建“稀疏”数组（spare array，即含有空白或空缺单元的数组）要特别注意：

Var a=[ ];

a[0]=1;

此处没有设置a[1]单元

a[2]=3;

a[1];//undefined

a.length;//3

1. 数组通过数字进行索引，但因为其也是对象，所以也可以包含字符串键值和属性（但这些并不包含在数组长度内）

Var a=[ ];

a[0]=1;

a[“foobar”]=2;

a.length;//1

1. 如果键值能够被强制类型转换为十进制数字的话，他就会被当作数字索引来处理。

Var a=[ ];

a[“13”]=42;

a.length;//14

建议使用对象来存放键值/属性值，用数组来存放数字索引值

1. 将类数组转换为真正的数组：通过数组工具函数（如indexof(),concat(),forEach()等）来实现
2. Array.from()
3. 字符串
4. 字符串不可变是指字符串的成员函数不会改变其原始值，而是创建并返回一个新的字符串。而数组的成员函数都是在其原始值上进行操作
5. 数组处理字符串很方便，因此我们可以通过“借用数组的非变更方法来处理字符串：例var c=Array.prototype.join.call(a,”-”);
6. 数组有一个字符串没有的变更成员函数reverse（），字符串不能“借用“数组的可变更成员函数，因为字符串是不可变的

可以先将字符串转换为数组，待处理完之后再转换回来(此方法不适用于包含复杂字符（如星号、多字节字符等）的字符串)：

Var c=a;

c.split(“”);

c.reverse();

c.join(“”);

c;

（4）可以使用join（）将字符串数组转换为字符串

1. 数字
2. 默认情况下大部分数字都以十进制显示，小数最后面的0被省略；特别大和特别小的数字默认用指数格式显示，与toExponential()函数的输出结果相同

数字值可以使用number对象进行封装，因此数字值可以调用number.prototype中的方法：例toFixed()方法可指定小数部分的显示位数,如果指定的小数部分的显示位数多余实际位数就用0补齐

Var a=42.59;

a.toFixed(0);//43

（2）toprecision()方法用来指定有效数位的显示位数：例

Var a=42.59;

a. toprecision(2);//43