

类型

JavaScript分基本数据类型和引用类型两大类：

- 基本数据类型：（数据存在栈内存中，按值拷贝）
 - undefined
 - null
 - number
 - boolean
 - string
 - symbol
- 复杂数据类型：（数据存在堆内存中，按引用拷贝）
 - Object
 - Function
 - Array
 - Date
 - Math
 - 基本包装类型（Number、Boolean、String、Symbol）
 - 自定义类型

类型转换

类型转换只有三种情况：

- 转number
- 转string
- 转boolean

原始值	转换目标	结果
number	布尔值	除了 0、-0、NaN 都为 true
string	布尔值	除了空串都为 true
undefined、null	布尔值	FALSE
引用类型	布尔值	TRUE
number	字符串	5 => '5'
Boolean、函数、Symbol	字符串	'true'
数组	字符串	[1,2] => '1,2'
对象	字符串	'[object Object]'
string	数字	'1' => 1, 'a' => NaN
数组	数字	空数组为0, 存在一个元素且为数字转数字, 其他情况 NaN
null	数字	0
除了数组的引用类型	数字	NaN
Symbol	数字	抛错

隐式转换规则：

1. 转成string类型：+（只要+运算符一端有string, 最终结果一定是string）
2. 转成number类型：
 - 自增自减运算符：++、--
 - 算数运算符：+、-、*、/、%
 - 关系运算符：>、<、>=、<=、!=、=、!==
3. 转成boolean类型：！（布尔运算符）

==比较运算的隐式转换规则：

1. 比较双方都是原始类型，则二者都转为Number进行比较
2. 比较双方中，一方是引用类型，另一方是Number，则引用类型转Number进行比较
3. 比较双方中，一方是引用类型，另一方是string，则引用类型转string进行比较
4. 比较双方中，一方是引用类型，另一方是boolean，则二者均转为数字进行比较
5. 比较双方都是引用类型，则比较的是内存地址

经典面试题：

[js面试题大坑——隐式类型转换](#)

