1. 博客
2. toString()
   1. undefined和null没有toString()
   2. 布尔值加引号返回

console.log(true.toString());//'true'

console.log(Boolean.toString());//'function Boolean() { [native code] }'

* 1. 字符串类型原值返回

console.log('abc'.toString());//'abc'

console.log(String.toString());//'function String() { [native code] }'

* 1. 数值类型

正浮点数及NaN、Infinity加引号返回

console.log(Number.toString());//'function Number() { [native code] }'

console.log(1.2.toString());//'1.2'

console.log(NaN.toString());//'NaN'

console.log(Infinity.toString());//'Infinity'

负浮点数及加’+’的浮点数直接跟上toString()，相当于先运行toString()，再添加正负号变成数字

console.log(+1.2.toString(),typeof +1.2.toString());//1.2 "number"

console.log(-1.2.toString(),typeof -1.2.toString());//-1.2 "number"

整数直接跟toString()会报错，整数后面的点会被识别为小数点，因此应该加小括号 toString()里面可以加参数，代表要转换的进制

//console.log(1.toString());//报错

console.log((1).toString());//"1"

console.log((9).toString(8));//"11"

* 1. 对象Object类型以及自定义类型返回[object Object]

console.log({}.toString());//'[object Object]'

console.log(({}).toString());//'[object Object]'

console.log({a:123}.toString());//'[object Object]'

console.log(Object.toString());//'unction Object() { [native code] }'

function Fn(){}

var obj=new Fn();

console.log(obj.toString());//'[object Object]'

* 1. 函数类型返回函数代码

console.log(Fn.toString());//'function Fn(){}'

console.log(Function.toString());//'function Function() { [native code] }'

* 1. 数组类型变成以逗号分隔的字符串

console.log([].toString());//''

console.log([1,2,3].toString());//'1,2,3'

console.log(Array.toString());//'function Array() { [native code] }'

* 1. 日期类型

console.log(new Date().toString());//'Wed Apr 11 2018 12:44:27 GMT+0800 (中国标准时间)'

console.log(Date.toString());//'function Date() { [native code] }'

* 1. Error类型

console.log(Error.toString());//'function Error() { [native code] }'

console.log(TypeError.toString());//'function TypeError() { [native code] }'

1. valueOf()
   1. 布尔值

console.log(true.valueOf());//true

console.log(Boolean.valueOf());//ƒ Boolean() { [native code] }

* 1. 字符串

console.log('abc'.valueOf());//'abc'

console.log(String.valueOf());//ƒ String() { [native code] }

* 1. 数字

整数要加小括号

console.log((0).valueOf());//0

console.log((-0).valueOf());//-0

浮点数原值返回

console.log(+1.2.valueOf());//1.2

console.log(-1.2.valueOf());//-1.2

-0的valueOf()是-0 -0的toString()是0

* 1. 对象

console.log({}.valueOf());//{}

console.log({a:123}.valueOf());//{a: 123}

console.log(Object.valueOf());//ƒ Object() { [native code] }

function Person(){this.name='cdx'}

var p1=new Person();

console.log(p1.valueOf());//Person {name: "cdx"}

* 1. 函数

console.log(Person.valueOf());//ƒ Person(){this.name='cdx'}

* 1. 数组

console.log([].valueOf());//[]

console.log([1,2,3].valueOf());//(3) [1, 2, 3]

console.log(Array.valueOf());//ƒ Array() { [native code] }

* 1. 日期

console.log(new Date().valueOf());//1523423207114

console.log(Date.valueOf());//ƒ Date() { [native code] }

* 1. Error

console.log(Error.valueOf());//ƒ Error() { [native code] }

console.log(TypeError.valueOf());//ƒ TypeError() { [native code] }

1. 表达式

原始表达式：字面量、关键字、变量

复杂表达式：原始表达式+操作符

1. 一元+

对数值没有影响，对非数值会调用Number()

1. 数字+非数字

如果一个操作数是对象，对象会转换为原始值：日期对象通过toString()，其他对象通过valueOf() ，如果valueOf()无法返回一个原始值则调用toString(),否则返回NaN

如果一个操作数是字符串，则将数字转换为字符串，进行字符串拼接操作

1. 数组方法

改变数组本身：pop splice sort

不改变数组：concat slice