

Số – Number

Không như nhiều ngôn ngữ lập trình khác, kiểu số thường chia ra làm nhiều loại như **integer**, **short**, **long**, **float**... Riêng Javascript chỉ có kiểu **number**

Số nguyên trong Javascript có độ chính xác đến **15** chữ số

```
var x = 999999999999999 // x sẽ là 999999999999999
var y = 999999999999999 // y sẽ là 1000000000000000
```

Giới hạn của số thập phân là **17** chữ số, nhưng không phải việc tính toán số thập phân lúc nào cũng chính xác 100%. Tìm hiểu [floating-point](#)

```
var x = 0.2 + 0.1 // x sẽ bằng 0.30000000000000004
```

Để giải quyết điều này, chúng ta nên nhân nó thành số nguyên rồi thực hiện

```
var x = (0.2 * 10 + 0.1 * 10) / 10 // x sẽ bằng 0.3
```

Hoặc làm tròn đến hàng thập phân cho phép

```
let a = 0.2 + 0.1
console.log(Number(a.toFixed(1))) // 0.3
```

Hoặc dùng các thư viện ngoài như [math.js](#)

```
math.add(math.fraction(0.1), math.fraction(0.2)) // 0.3
```

Nếu cộng 2 số thì kết quả sẽ là số. Nhưng nếu chuỗi cộng với số thì kết quả là chuỗi. Vậy nên khi tính toán thì nên chuyển hết về dạng số

```
var x = '10'
var y = 20
var z = x + y // z sẽ là 1020 (một string)
```

NaN – Not a Number

NaN là một giá trị trong Javascript dùng để xác định một số không phải là một số hợp lệ

```
var x = 100 / 'Apple' // x sẽ là NaN (Not a Number)
typeof NaN // returns "number"
// chúng ta có thể dùng function isNaN() để kiểm tra giá trị có phải là NaN hay không
isNaN(x) // return true
```

Infinity – Dương vô cực

Infinity nghĩ là dương vô cực, **-Infinity** nghĩa là âm vô cực

```
var x = 2 / 0 // x sẽ là Infinity
var y = -2 / 0 // y sẽ là -Infinity
typeof Infinity // returns "number"
```

Hexadecimal – Hệ thập lục phân

Nếu bạn biểu diễn bắt đầu bằng 0x thì JS sẽ hiểu đây là hệ thập lục phân

```
var x = 0xff // x sẽ là 255
```

Một số phương thức hay dùng

toString(): Ép kiểu số về chuỗi

```
var x = 123
x.toString()(
  // '123'
  123
).toString() // '123'
```

toFixed(): Làm tròn

```
var x = 9.656
x.toFixed(0) // returns 10
x.toFixed(2) // returns 9.66
x.toFixed(4) // returns 9.6560
x.toFixed(6) // returns 9.656000
```

Ép kiểu về số

Chúng ta có 3 cách

- **Number()** chuyển đổi giá trị về kiểu số
- **parseInt()** chuyển đổi giá trị về kiểu số nguyên
- **parseFloat()** chuyển đổi giá trị về kiểu số thập phân

```
Number(true) // returns 1
Number(false) // returns 0
Number('10') // returns 10
Number(' 10') // returns 10
Number('10 ') // returns 10
Number(' 10 ') // returns 10
Number('10.33') // returns 10.33
Number('10,33') // returns NaN
Number('10 33') // returns NaN
Number('John') // returns NaN
```

```
parseInt('10') // returns 10
parseInt('10.33') // returns 10
parseInt('10 20 30') // returns 10
parseInt('10 years') // returns 10
parseInt('years 10') // returns NaN
parseFloat('10') // returns 10
parseFloat('10.33') // returns 10.33
parseFloat('10 20 30') // returns 10
parseFloat('10 years') // returns 10
parseFloat('years 10') // returns NaN
```