

Number

1. Giới thiệu về kiểu number
2. Object Number có gì?
3. Object Math có gì?
4. Imprecise calculations
5. Rounding - Làm tròn số
6. Random number
7. Bài tập thực hành

1. Giới thiệu về kiểu number:

Có 2 loại số

- Integer (int): số nguyên (1, 2, 3, ...)
- Float (float): số thực (1.5, 2.5, 3.7, ...). Dùng dấu chấm để ngăn cách phần thập phân.

Lưu ý: Javascript chỉ có một kiểu dữ liệu là number dùng cho cả số nguyên và số thực

Number trong JS được lưu dưới dạng double-precision 64-bit binary format [IEEE 754](#).

```
const count = 1;
const mark = 9.5;
const long = 1000000;
const longer = 1_000_000; // rare usage
const short = 1e6; // usually see this
const smaller = 0.0001;
const smallerShort = 1e-4; // rối não
```

Arithmetic operators

```
const sum = 1 + 2; // 3
const subtract = 1 - 2; // -1
const multiply = 1 * 2; // 2
const divide = 1 / 4; // 0.25
const remainder = 7 % 5; // 2 because 7 = 5x1 + 2 (remainder)
// increment
let count = 1;
const n = count++; // 1 postfix, return first, then increase
let count = 1;
const n = ++count; // 2 prefix, increase first, then return
// decrement
let count = 1;
const n = count--; // 1 postfix, return first, then decrease
let count = 1;
const n = --count; // 0 prefix, decrease first, then return
// unary plus: attempts to convert to number if it's not a number
```

```
const n = +'123'; // 123
// unary negation: return the negation of the operand
const n = -'123'; // -123
// exponentiation operator
const n = 2 ** 3; // means 2^3 = 8
```

Assignment operators

```
// addition assignment
let n = 1;
n += 1; // 2, similar to n = n + 1
// subtraction assignment
let n = 1;
n -= 1; // 0 similar to n = n - 1
// multiplication assignment
let n = 1;
n *= 3; // 3 similar to n = n * 3
// division assignment
let n = 1;
n /= 2; // 0.5 similar to n = n / 2
```

toString(base)

```
const n = 20;
n.toString(); // '20' (default is 10)
n.toString(10); // '20' Decimal
n.toString(16); // '14' Hexadecimal
n.toString(8); // '24' Octal
n.toString(2); // '10100' Binary
```

Tham khảo:

- Cách mà number trong javascript được lưu trữ: <http://steve.hollasch.net/cgindex/coding/ieeefloat.html>
- <https://javascript.info/number>
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number
- [Full list of operators](#)

2. Number - A primitive wrapper object:

Use Number as a function

When used as a function, `Number(value)` converts a string or other value to the `Number` type. If the value can't be converted, it returns `NaN`.

```
Number(123); // 123
Number('123'); // 123
Number('abc'); // NaN
```

Static properties

#	Prop	Value	Desc
1	Number.EPSILON	2.220446049250313e-16	Độ chênh lệch nhỏ nhất giữa 2 số
2	Number.MIN_VALUE	5e-324	Số dương nhỏ nhất (gần tới số 0)
3	Number.MAX_VALUE	1.7976931348623157e+308	Số dương lớn nhất
4	Number.MIN_SAFE_INTEGER	-9007199254740991	2 ⁵³ - 1
5	Number.MAX_SAFE_INTEGER	9007199254740991	2 ⁵³ - 1
6	Number.NaN	NaN	Not a Number
7	Number.NEGATIVE_INFINITY	-Infinity	Âm vô cùng
8	Number.POSITIVE_INFINITY	Infinity	Dương vô cùng

Tham khảo: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number#static_properties

Phân biệt toFixed() và toPrecision():

- toFixed() và toPrecision() đều chuyển từ number thành string
- toFixed(digits) thì cố định số lượng digits sau dấu chấm.
- toPrecision(digits) thì làm tròn tới digits số có nghĩa.

```
const n = 123.525;
n.toFixed(); // 124 as default value for arg is 0
n.toFixed(0); // 124
n.toFixed(1); // 123.5
n.toFixed(2); // 123.53
n.toFixed(3); // 123.525
n.toFixed(4); // 123.5250
n.toFixed(5); // 123.52500
```

```
const n = 123.525;
n.toPrecision(); // 123.525 similar to n.toString()
n.toString(); // 123.525
n.toPrecision(0); // error argument must be between 1 and 100
n.toPrecision(1); // 100
n.toPrecision(2); // 120
n.toPrecision(3); // 124
n.toPrecision(4); // 123.5
n.toPrecision(5); // 123.53
n.toPrecision(6); // 123.525
n.toPrecision(7); // 123.5250
n.toPrecision(8); // 123.52500
```

3. Math - Built-in object for mathematical constants and functions:

Math is a built-in object that has properties and methods for mathematical constants and functions. It's not a function object.

Các props phổ biến

#	Prop	Desc
1	Math.PI	Giá trị của PI, khoảng 3.14159
2	Math.SQRT2	Giá trị của căn bậc 2 của 2 ~ 1.414

Các methods phổ biến

#	Method	Desc
1	Math.ceil(x)	Làm tròn lên số nguyên gần nhất
2	Math.floor(x)	Làm tròn xuống số nguyên gần nhất
3	Math.round(x)	Làm tròn tới số nguyên gần nhất
4	Math.trunc(x)	Hàm lấy phần nguyên, bỏ phần thập phân
5	Math.random()	Random số thực từ 0 --> 1
6	Math.abs(x)	Lấy giá trị tuyệt đối
7	Math.pow(x, y)	Hàm lũy thừa, x^y
8	Math.sqrt(x)	Hàm căn bậc 2

```
Math.abs(-5); // 5
Math.abs(5); // 5
Math.pow(2, 3); // 8
Math.pow(10, 2); // 100
Math.sqrt(16); // 4
Math.sqrt(81); // 9
```

Tham khảo: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math

4. Imprecise calculations:

```
0.3 === 0.1 + 0.2; // true or false ???
```

```

console.log((0.1).toFixed(20));
console.log((0.2).toFixed(20));
console.log((0.1 + 0.2).toFixed(20));
console.log((0.3).toFixed(20));
console.log(0.1 + 0.2 == 0.3);
console.log(0.1 + 0.2 === 0.3);
console.log(Math.abs(0.1 + 0.2 - 0.3) < Number.EPSILON);
console.log((0.1 + 0.2).toFixed(1) === (0.3).toFixed(1));

```

Lưu ý:

- Không phải số nào cũng được lưu chính xác trong Javascript (thực ra việc này cũng xảy ra ở những ngôn ngữ khác do dùng chung chuẩn lưu dữ liệu)
- Khi làm việc với số thực cần lưu về vấn đề này để xử lý cho đúng.
- So sánh so thực bằng nhau thì dùng hằng số EPSILON hoặc toFixed() để làm tròn số trước khi so sánh nhé.

Tham khảo: <https://javascript.info/number#imprecise-calculations>

5. Rounding - Làm tròn số:

4 hàm làm tròn số:

1. Math.ceil(x) Làm tròn lên số nguyên gần nhất.
2. Math.floor(x) Làm tròn xuống số nguyên gần nhất.
3. Math.round(x) Làm tròn tới số nguyên gần nhất.
4. Math.trunc(x) Hàm lấy phần nguyên, bỏ phần thập phân.

Bảng so sánh các hàm:

Value	Math.floor	Math.ceil	Math.round	Math.trunc
3.2	3	4	3	3
2.5	2	3	3	2
-1.1	-2	-1	-1	-1
-1.6	-2	-1	-2	-1

Tham khảo: <https://javascript.info/number#rounding>

6. Random number in JS:

1. Random một số ngẫu nhiên trong khoảng [0, n]
2. Random một số ngẫu nhiên trong khoảng [a, b] với $a < b$

Gợi ý sử dụng hàm `Math.random()` để làm việc này, nhưng hàm này chỉ trả về trong khoảng [0, 1]

```
function randomNumber(n) {  
  if (n <= 0) return -1;  
  const random = Math.random() * n;  
  return Math.round(random);  
}  
console.log(randomNumber(3));  
console.log(randomNumber(3));  
console.log(randomNumber(3));  
console.log(randomNumber(3));  
console.log(randomNumber(3));  
console.log(randomNumber(3));  
console.log(randomNumber(3));  
console.log(randomNumber(3));  
console.log(randomNumber(3));  
console.log(randomNumber(3));  
console.log(randomNumber(3));  
console.log(randomNumber(-1));  
console.log(randomNumber(100));
```

```
// 2. Random a number in range of [a, b] với a < b  
// min: a  
// range: b - a  
// [10, 100]  
// 10  
// 100 - 10 = 90  
function randomNumberInRange(a, b) {  
  if (a >= b) return -1;  
  const random = Math.random() * (b - a);  
  return Math.round(random) + a;  
}  
console.log(randomNumberInRange(10, 99));  
console.log(randomNumberInRange(10, 99));  
console.log(randomNumberInRange(10, 99));  
console.log(randomNumberInRange(100, 999));  
console.log(randomNumberInRange(100, 999));  
console.log(randomNumberInRange(100, 999));  
console.log(randomNumberInRange(1000, 9999));  
console.log(randomNumberInRange(1000, 9999));  
console.log(randomNumberInRange(1000, 9999));  
console.log(randomNumberInRange(1000, 9999));
```

7. Bài tập thực hành:

Bài tập toán học:

```
// 1. Write a function to calculate the area of a rectangle  
function calcAreaOfRectangle(a, b) {
```

```
if (a <= 0 || b <= 0) return -1;
return a * b;
}
console.log(calcAreaOfRectangle(0, 1));
console.log(calcAreaOfRectangle(4, 2));
```

```
// 2. Write a function to calculate perimeter of a rectangle
function calcPerimeterOfRectangle(a, b) {
  if (a <= 0 || b <= 0) return -1;
  return (a + b) * 2;
}
console.log(calcPerimeterOfRectangle(4, 2));
```

```
// 2. Write a function to calculate perimeter of a rectangle
function calcPerimeterOfRectangle(a, b) {
  if (a <= 0 || b <= 0) return -1;
  return (a + b) * 2;
}
console.log(calcPerimeterOfRectangle(4, 2));
```

```
// 3. Write a function to calculate the area of a circle
// S = PI * R * R
function calcAreaOfCircle(r) {
  if (r <= 0) return -1;
  return Math.PI * r * r;
}
console.log(calcAreaOfCircle(5));
```

Bài tập number - dạng kiểm tra đúng sai

```
// 1. Write a function to check if a number is odd/even
function isEven(n) {
  return n % 2 === 0;
}
function isOdd(n) {
  return n % 2 === 1;
}
```

```
// 2. Write a function to check if a number is divisible by 5
function isDivisibleBy5(n) {
  return n % 5 === 0;
}
```

```
// 3. Write a function to check if a number is perfect square
// n = a * a
// RECOMMENDED
function isPerfectSquare(n) {
  if (n <= 0) return false;
  const sqrtN = Math.sqrt(n);
  const sqrtNInt = Math.trunc(sqrtN);
  return sqrtNInt * sqrtNInt === n;
}
// 25
// 5
// sqrtNInt = 5
// 20
// 4.x
// sqrtNInt = 4
console.log(isPerfectSquare(20));
console.log(isPerfectSquare(25));
console.log(isPerfectSquare(16));
console.log(isPerfectSquare(4));
console.log(isPerfectSquare(9));
console.log(isPerfectSquare(10));
```

Bài tập về number - Tìm max

```
// 1. Convert hours to seconds
function convertHoursToSeconds(hours) {
  if (hours < 0) return -1;
  if (hours === 0) return 0;
  const SECONDS_PER_HOUR = 3600;
  return hours * SECONDS_PER_HOUR;
}
```

```
// 2. Given 3 numbers, find max
function findMax(a, b, c) {
  let max = a;
  if (b > max) max = b;
  if (c > max) max = c;
  return max;
}
console.log(findMax(7, 5, 3));
```

```
// 3. Given 3 numbers, find max even number
function findMaxEven(a, b, c) {
  let max = Number.NEGATIVE_INFINITY;
```



```
if (a % 2 === 0 && a > max) max = a;
if (b % 2 === 0 && b > max) max = b;
if (c % 2 === 0 && c > max) max = c;
return max;
}
console.log(findMaxEven(4, 5, 8));
```

Bài tập number - Rút trích chữ số

```
// 1. Get the ones of a number having 3 digits
function extractTheOnes(n) {
  if (n.toString().length !== 3) return -1;
  // 123 % 10 = 3
  return n % 10;
}
console.log(extractTheOnes(123));
console.log(extractTheOnes(1234));
```

```
// 2. Get the tens of a number having 3 digits
function extractTheTens(n) {
  if (n.toString().length !== 3) return -1;
  // 223 % 100
  // 23
  // 23 / 10 = 2.3 --> 2
  return Math.trunc((n % 100) / 10);
}
console.log(extractTheTens(223));
```

```
// 3. Get the hundreds of a number having 3 digits
function extractTheHundreds(n) {
  if (n.toString().length !== 3) return -1;
  // 523 / 100 = 5.23 --> 5
  return Math.trunc(n / 100);
}
console.log(extractTheHundreds(523));
```