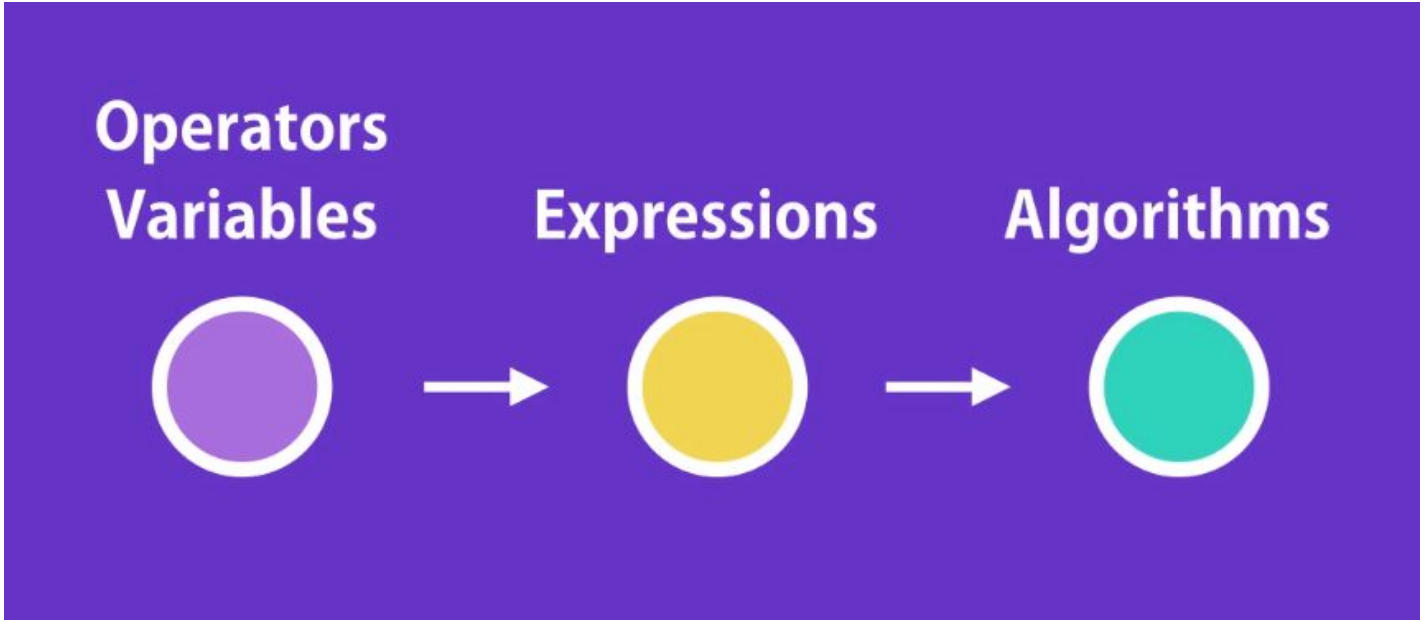
Operators

Ba Nguyễn

TechMaster
Học là có việc

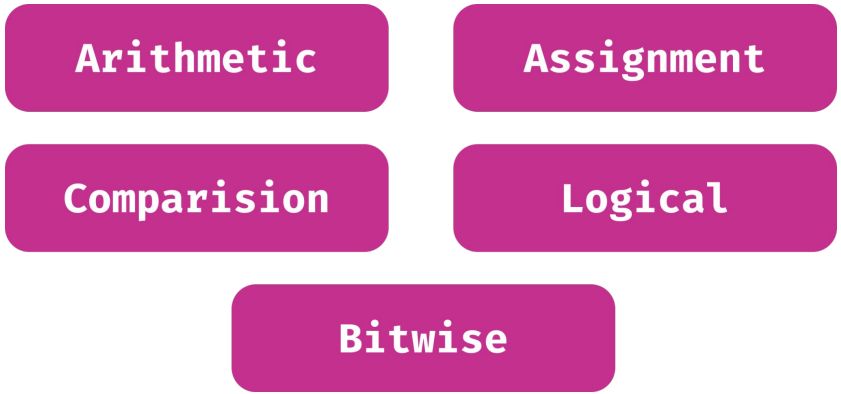
Operators



TechMaster
Học là có việc

Operators

Toán tử trong JavaScript được chia thành 5 loại:



💡 Quy tắc thực hiện biểu thức theo thứ tự *từ trái qua phải, dựa theo độ ưu tiên toán tử*. Các toán tử có độ ưu tiên khác nhau, quyết định phép tính nào sẽ được thực hiện trước

💡 Tham khảo độ ưu tiên toán tử: [mdn/operators_precedence]

TechMaster
Học là có việc

Arithmetic Operators

Đối với kiểu dữ liệu `string` phép `+` chuyển đổi kiểu dữ liệu của toán hạng về kiểu `string` và thực hiện nối chuỗi

```
1 + "2"; // "12"
1 + 2 + "3"; // "33"
"1" + 2 + true; // "12true"
```

Phép nối chuỗi ***chỉ hoạt động duy nhất với toán tử*** `+` , với những toán tử số học khác, mọi kiểu dữ liệu được chuyển về `number`

```
6 - "2" + null - true; // 3
"5" / 3 + 2 / "1"; // 4
```

TechMaster
Học là có việc

Arithmetic Operators

Basic operator: `+ - * / % **`

```
// Chia lấy phần dư %
5 % 3; // 2
"6" % 2; // 0
1 % true; // 0

// Lũy thừa **
2 ** 2; // 4
2 ** "2"; // 4
2 ** false; // 1

// Mọi phép tính (trừ phép + chuỗi) với NaN đều cho kết quả là NaN
5 - NaN / 2; // NaN
1 * 2 * undefined; // NaN
```

TechMaster
Học là có việc

Exercise

1. Viết chương trình nhập 2 số `a`, `b`, tính và in ra kết quả tất cả phép tính cơ bản với 2 số đó
2. Viết chương trình nhập vào chiều dài, chiều rộng của hình chữ nhật, tính và in ra chu vi, diện tích của hình chữ nhật đó
3. Viết chương trình nhập vào bán kính hình tròn, tính và in ra chu vi, diện tích của hình tròn

TechMaster
Học là có việc

Assignment Operators

```
// Gán =
let a = 3; // a = 3
let b = 1 + 2 + 3; // b = 6
let c = (a = b + 5); // b = 6, a = 11, c = 11

// Gán kết hợp
a += b; // a = a + b
a /= b; // a = a / b
a %= b + 5; // a = a % (b + 5)
a *= a + 2; // a = a * (a + 2);
```

TechMaster
Học là có việc

Asignment Operators

Increment, Decrement (tự tăng/giảm)

`++` và `--` là hai toán tử đặc biệt, nó thực hiện phép tính ***tăng/giảm*** giá trị của biến đi `1`, hai toán tử này có thể đặt ở trước biến - *prefix* hoặc sau biến - *postfix*.

Khi đặt trên một câu lệnh riêng biệt thì không có sự khác nhau

```
let a = 1;
a++; // a = 2
a--; // a = 1
++a; // a = 2
--a; // a = 1
```

TechMaster
Học là có việc

Asignment Operators

Increment, Decrement (tự tăng/giảm)

Tuy nhiên, khi đặt trong một biểu thức, *postfix* - tăng/giảm `a` đi `1` và trả về ***giá trị trước đó (giá trị trước khi tăng/giảm)***, *prefix* - cũng tăng/giảm `a` đi `1` nhưng trả về ***giá trị mới (giá trị sau khi tăng/giảm)***

```
let a = 1;
let b = a++ + 2; // a = 2, b = 1 + 2 = 3
let c = ++a + 2; // a = 3, c = 3 + 2 = 5

let d = a++ + ++a - a-- - --a;
// a = 3, d = 3 + 5 - 5 - 3 = 0
```

TechMaster
Học là có việc

Exercise

Tính giá trị các biểu thức

```
let a = 1,
    b = (a % 2) * 2,
    c = a++ - b--,
    d = "0";

a + b + c + d;
a - b + c - d;
a-- + (b-- * c) / d;
++a - +b * c + d;
d + ++a + (--b % c);
a-- - +d++ - ++c + b--;
a++ - b-- + ++c + --d;
```

```
let a = 1,
    b = (a * 2) / 2,
    c = a-- + b++,
    d = "-0";

a - b - c - d;
a + b - c + d;
a++ - (b++ / c) * d;
--a + -b / c - d;
d - --a - ++b * c;
a++ + -d-- + --c - b++;
a-- + b++ - --c - ++d;
```

TechMaster
Học là có việc

Comparision Operators

Toán tử so sánh `==` `!=` `>` `>=` `<` `<=` `===` `!==`

💡 Kết quả của các phép so sánh là một giá trị `boolean`

`==` `!=` `>` `>=` `<` `<=` tự động chuyển đổi kiểu dữ liệu của 2 toán hạng về cùng một kiểu và thực hiện so sánh

```
2 < 3; // true
2 >= 3; // false
2 == "2"; // true
2 != "2"; // false
1 > null; // null → 0 → true
2 <= "2"; // true
```

TechMaster
Học là có việc

Comparision Operators

Toán tử so sánh `==` `!=` `>` `>=` `<` `<=` `===` `!==`

`===` và `!==` (*strict comparison*) so sánh cả kiểu giá trị của dữ liệu

```
2 === 2; // number vs number và 2 == 2 → true
2 === "2"; // number vs string → false
2 !== "2"; // number vs string → true
2 !== 2; // number vs number → false
0 === null; // number vs object → false
```

TechMaster
Học là có việc

Comparision Operators

So sánh chuỗi

JavaScript sử dụng bảng mã Unicode, khi so sánh 2 chuỗi, nó thực hiện so sánh từng ký tự dựa theo thứ tự trong bảng mã (Unicode table)

```
"a" > "A"; // true
"A" > "Z"; // false
"Ba" == "Ba"; // true
"Ba" ≤ "Ba Nguyễn"; // true
```

💡 Tham khảo bảng mã Unicode: wiki/unicode_character

TechMaster
Học là có việc

Comparision Operators

`null`, `undefined`, `NaN`

`null` và `undefined` là 2 trường hợp đặc biệt

```
null == 0; // false
null ≤ 0; // true
null ≥ 0; // true
null == undefined; // true
null === undefined; // false
```

💡 Mọi biểu thức so sánh với `NaN` đều cho kết quả là `false`

TechMaster
Học là có việc

Logical Operators

Toán tử logic `||` - or, `&&` - and, `!` - not

`||` - `or` tìm giá trị `true` đầu tiên trong biểu thức (chuyển về kiểu `boolean`) và trả về giá trị đó, nếu không có giá trị nào là `true` thì trả về giá trị cuối cùng trong biểu thức (bất kể `true` hay `false`)

```
true || false; // true
0 || 1; // 1
0 || false || ""; // ""
"abc" || true; // "abc"
let age = 16;
age > 18 || alert("Bạn chưa đủ tuổi"); // ?
```

TechMaster
Học là có việc

Logical Operators

Toán tử logic `||` - or, `&&` - and, `!` - not

`&&` - `and` tìm giá trị `false` đầu tiên trong biểu thức (chuyển về kiểu `boolean`) và trả về giá trị đó, nếu không có giá trị nào là `false` thì trả về giá trị cuối cùng trong biểu thức (bất kể `true` hay `false`)

```
true && false; // false
0 && 1; // 0
0 && false && ""; // 0
"abc" && true; // true
let age = 16;
age > 18 && alert("Bạn đủ tuổi rồi :)"); // ?
```

TechMaster
Học là có việc

Logical Operators

Toán tử logic `||` - or, `&&` - and, `!` - not

`!` - `not` chuyển giá trị về kiểu `boolean` và phủ định nó

```
!""; // true
!"123"; // false
!!false; // false
!!"xxx"; // true
!!!!!!!!!!!!!!false; // false
```

TechMaster
Học là có việc

Exercise

Tính giá trị các biểu thức

```
let a = true,
    b = !a;
let c = (!a && !!b) || 0;

a && b && c;
a || b || c;
(a && !b) || !!c;
!(a || !b) && c;
!!(a && !!b) || !c;
```

```
let a = false,
    b = !!a;
let c = a || (!b && 0);

a && b && c;
a || b || c;
!a || (b && !c);
!!(!a && b) || c;
!(!a || !b) || c;
```

TechMaster
Học là có việc

Exercises

1. Viết chương trình nhập một số `n`, kiểm tra và in ra số đó là chẵn hay lẻ, sử dụng toán tử logic
2. Viết chương trình nhập vào 2 số `a`, `b`. Kiểm tra và in ra số lớn hơn, sử dụng toán tử logic
3. Viết chương trình nhập một chuỗi str. Sử dụng toán tử logic kiểm tra và in ra nếu:
   - `str = Ba` → Thầy Ba đẹp trai
   - Các trường hợp khác → Thầy Ba vẫn đẹp trai :)

TechMaster
Học là có việc

Homework

1. Viết chương trình nhập vào một giá trị đo độ dài ở đơn vị *cm*, tính và in ra giá trị tương ứng ở các đơn vị *mm*, *m*, *km*
2. Viết chương trình nhập vào một giá trị nhiệt độ ở đơn vị *Celsius*, in ra nhiệt độ ở đơn vị *Fahrenheit* và *Kevin* tương ứng
3. Viết chương trình nhập giá trị thời gian *tính theo số giây*, tính và in ra giá trị giờ/phút/giây tương ứng theo định dạng `h:m:s`
4. Viết chương trình nhập hệ số `a`, `b` của phương trình bậc 1 `ax + b = 0`, tính và in ra nghiệm của phương trình. Sử dụng toán tử logic để đánh giá các trường hợp của phương trình
5. Viết chương trình nhập 3 số `a`, `b`, `c`. Sử dụng toán tử logic kiểm tra và in ra số lớn nhất
6. Viết chương trình nhập 3 số `a`, `b`, `c` bất kỳ. Kiểm tra 3 số đó có tạo thành tam giác hợp lệ hay không và in ra kết quả, sử dụng toán tử logic

TechMaster
Học là có việc