

JavaScript Operators

Arithmetic Operators

```
let x = 10;
```

```
let y = 3;
```

```
console.log(x + y);
```

```
console.log(x - y);
```

```
console.log(x * y);
```

```
console.log(x / y);
```

```
console.log(x % y);
```

```
console.log(x ** y); (exponential)
```

```
console.log(++x); (increment)
```

```
console.log(--x); (decrement)
```

Assignment Operators

```
let x = 10;
```

```
x + 1;
```

```
x = x + 1;
```

```
x += 5;
```

```
x *= 3;
```

Comparison Operators

Relational operators

let x = 1;

console.log(x > 0);

Result: True

console.log(x >= 1);

console.log(x < 1);

console.log(x <= 1);

Equality operators

Strict Equality (Type + Value)

console.log(x === 1); True

console.log(x !== 1); False

Loose Equality

console.log(1 == 1); True

console.log('1' == 1); True

(Automatically convert the type)

Strict Equality Operator (Type + Value)

console.log(1 === 1); True

console.log('1' === 1); False

Ternary Operator / Conditional Operator

let points = 110;

let type = points > 100 ? 'Gold' : 'Silver';

console.log(type);

Result: Gold

Logical Operators

Logical AND (&&)

```
Console.log(true && false);
```

Returns true if both operands are true

Logical OR (||)

```
let highIncome = true;
```

```
let goodCreditScore = true;
```

```
let eligibleForLoan = highIncome || goodCreditScore;
```

```
Console.log(eligibleForLoan);
```

Result: true.

Logical NOT (!)

```
let highIncome = true;
```

```
let goodCreditScore = true;
```

```
let eligibleForLoan = highIncome || goodCreditScore;
```

```
Console.log('eligible; eligible for loan');
```

```
let applicationRefused = !eligibleForLoan;
```

```
Console.log('Application Refused',  
applicationRefused);
```

Bitwise Operators

1 = 00 00 0001

2 = 00 00 0010

3 = 00 000011

Console.log(1 | 2); // Bitwise OR result = 00000011

Console.log(1 & 2); // Bitwise AND result = 00000000

Control Flow

Conditional Statements

(i) if...else

Syntax

if (condition) {

Statement

}

else if (another condition) {

Statement

}

else

Statement

Q ⇒ // Hours
// If hour is between 6am and 12pm: Good morning
// If it is between 12pm and 6pm: Good afternoon
// Otherwise: Good evening!

```
let hour = 18;  
if ( hour >= 6 && hour < 12 )  
    console.log('Good morning');  
  
else if ( hour >= 12 && hour < 18 )  
    console.log('Good afternoon Evening');  
  
else  
    console.log('Good Evening');
```

(2) Switch

Example:

```
let role;  
Switch (role) {  
    Case 'Guest':  
        console.log('Guest User');  
        break;  
  
    Case 'moderator':  
        console.log('Moderator User');  
        break;  
}
```

default:

```
console.log('Unknown User');
```

}

Loop

Repeat ~~Use~~ ~~Repeat~~ an action for number of times

(1) For loop

example:

```
for (let i = 1; i <= 5; i++) {  
  if (i % 2 !== 0) console.log(i);  
}
```

(2) while loop

example:

```
let i = 0;  
while (i <= 5) {  
  if (i % 2 !== 0) console.log(i);  
  i++;  
}
```

3) do while loop

example:

```
let i = 0;  
do {
```

3) do while loop

example:

```
let i = 9;
```

```
do {
```

```
  if (i % 2 !== 0) console.log(i);
```

```
  i++;
```

```
} while (i <= 5);
```

4) for...in loop

iterate over object

```
const person = {
```

```
  name: 'mosh',
```

```
  age: 30
```

```
};
```

```
for (let key in person)
```

```
  console.log(key);
```

result: name

~~age~~ age

```
console.log(key, person[key]);
```

example:

name mosh

age 30

iterate over array

```
const colors = ['red', 'blue', 'green'];
```

```
for (let index in colors)
```

```
  console.log(color index, colors[index]);
```

5) for of loop

for of loop is used to iterate over arrays.

example

```
const colors = ['Red', 'Blue', 'Green'];
```

```
console.log
```

```
for (let color of colors)
```

```
  console.log(color);
```

result: ~~red~~ Red
Blue
Green

Break and Continue

Break:

Break is used to jump to outside from the block

```
example: let i = 0;  
while (i <= 10) {  
  if (i === 5) break;  
  console.log(i);  
  i++;  
}
```


Continue

Continue is used to jump beginning of the ~~for~~ block.

example: let i = 0

```
while (i <= 10) {
```

```
  if (i % 2 === 0) continue;
```

```
  console.log(i);
```

```
  i++;
```

```
}
```

In Conditional Operator boolean value is automatical assigns.

example:

```
console.log (islandscape (800, 600));
```

```
function islandscape (width, height) {
```

```
  return (width > height);
```

```
}
```

Result: False