1. Var swap

2. Split string into Array

3. the spread operator (three dots): **…**

1. swap var a and b:

(src: https://dmitripavlutin.com/swap-variables-javascript/)

1.1 Using temporary var:

VD:

let tmp = a;

a = b;

b = tmp;

The swapping of variables using a temporary variable works with any value type, like numbers, strings, booleans, objects.The downside of this approach is the need for a specialized temporary variable, plus the swapping happens in 3 statements.

1.2 Destructuring assignment:

[c, d] = [1, 2, 3]; // c = 1; d = 2

this line code is a destructuring assignment that destructures [1, 2, 3] array.

VD:

so on, we have:

[a, b] = [b, a];

It works with any data type: numbers, strings, booleans, objects.

1.3 Addition and difference:

a = a + b;

b = a – b;

a = a – b;

or

a = b + a – (b = a);

First, you can swap integers only. Secondly, be aware of the numbers overflow when performing the addition at the step a = a+b (the sum must be lower than Number.MAX\_SAFE\_INTERGER).

1.4 Bitwise Xor operator:

|  |  |  |
| --- | --- | --- |
| a | b | a^b |
| 0  1  0  1 | 0  1  1  0 | 0  0  1  1 |

Bitwise Xor(^) has 2 interesting properties:

n^n = 0

n^0 =n

Swapping variables using bitwise XOR operator has limitations: you can swap only integers.

a = a ^ b ;

b = a ^ b;

a = a ^ b;

Here’s an explanation why the swapping works:

1. a = a ^ b
2. b = a ^ b. Based on 1 a is substituted with a ^ b. Thus b = (a ^ b) ^ b = a ^ (b ^ b) = a ^ 0 = a. Remember than b is now a.
3. a = a ^ b. Based on 1 a is substituted with a ^ b and based on 2 b is substituted with a. Thus a = (a ^ b) ^ a = b ^ (a ^ a) = b ^ 0 = b. The variable a becomes b.

2.

Syntax:

<str\_name>.split(<separator>, <limit>)

Ex:

const s = “Hello beauty there”;

s.split(‘ ‘); // [‘Hello’, ‘beauty’, ‘there’]