



# JS LOOPS COURSE

APRIL 2022

YONATAN BENEZRA

- DO NOT DISTRIBUTE –

- DO NOT COPY – ALL RIGHTS RESERVED TO THE  
AUTHOR –

The C-style “Three-Expression” For Loops

Syntax:

```
const a = [1,2,3,4]
for (const i=0; i<a.length; i++){
  console.log(a[i])
}
```

## The “for...in...” Loops

```
var author = {name: "Yonatan", job: "programmer"}
for (var k in author){
  console.log(k + ':' + author[k]);
}
//name:Yonatan
//job:programmer
```

I recommend never using the above syntax for arrays.

Since arrays are in fact also objects, and the index of each item is regarded as the property of the object, the index order is implementation-dependent, and array values may not be accessed in the order you expect.

Therefore, applying “for...in...” to arrays may cause some unexpected results.

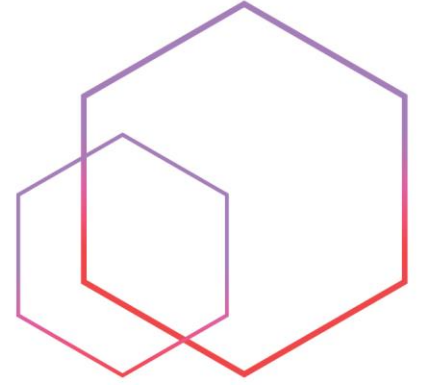
The following is a very good example of an answer to the question “Why is using “for...in” for array iteration a bad idea?” on StackOverflow.

```
// Somewhere deep in your JavaScript Library...
Array.prototype.foo = 1;

// Now you have no idea what the below code will do.
var a = [1, 2, 3, 4, 5];
for (var x in a){
  // Now foo is a part of EVERY array and
  // will show up here as a value of 'x'.
  console.log(x);
}
```

```
/* Will display:
0
```

```
1
2
3
4
foo
*/
```



Therefore, to make your code bug-free, never apply “for...in...” to arrays.

We could use the “forEach” Method for this

## The “for...of...” Syntax From ES6

For example, we can iterate the items of an array, as shown in the following example:

```
let author = [ 'Y', 'o', 'n', 'a', 't', 'a', 'n', ];
for (let char of author){
  console.log(char);
}
// Y
// o
// n
// a
// t
// a
// n
```

## The While Loop

The while loop loops through a block of code as long as a specified condition is true.

Syntax:

```
while (condition) {
  // code block to be executed
}
```

### Example:

In the following example, the code in the loop will run over and over again, as long as a variable (i) is less than 10:

```
while (i < 10) {  
  text += "The number is " + i;  
  i++;  
}
```

If you forget to increase the variable used in the condition, the loop will never end, which will cause your browser to crash

### The Do While Loop

The do while loop is a variant of the while loop. This loop will execute the code block once before checking if the condition is true, then it will repeat the loop as long as the condition is true.

Syntax:

```
do {  
  // code block to be executed  
}  
while (condition);
```

### Example:

The below example uses a do while loop. The loop will always be executed at least once, even if the condition is false, because the code block is executed before the condition is tested.

```
do {  
  text += "The number is " + i;  
  i++;  
}  
while (i < 10);
```

Do not forget to increase the variable used in the condition, otherwise the loop will never end!