

For loop

```
let iterable = [1, 2, 3];
for(let value of iterable) { console.log(value);
}
```

Slice VS Splice

slice ⇌ shallow

splice ✂ changes

The **slice()** method returns a **shallow** copy of a portion of an array into a new array object:

```
arr.slice([begin[, end]])
```

The **splice** method **changes** the content of an array by removing existing elements and/or adding new elements.

```
arr.splice(start, deleteCount [, item1 [, item2 ...]])
```

Chomp string

```
string = string.slice(0, -1);
```

Strings: substr VS substring

```
string.substr(start, length)
```

```
string.substring(start, stop)
```

Cloning/Assignment

```
Object.assign(obj1, obj2);
```

to clone:

```
let clone = Object.assign({}, obj1); // Though
not sure if shallow or deep copy
```

"Falsy" values in JS

Undefined

null

false

+0, -0, and NaN

" "

Array aggregation

```
array1 = array1.concat(array2);
```

Array includes element

```
array1.includes('xyz')
```

Case insensitive comparison

```
(a, b) =>
```

```
a.toLowerCase().localeCompare(b.to
```

Logging companions

→ <http://1lineart.kulaone.com/#/>

→ <http://bada55.io/>

For something like:

```
console.log('%cd[ o_0 ]b %s', 'color:
#fa5732;', 'DATA');
```

(always) Thanks to MDN



Always thanks to: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>



By **Dimitrios Mistriotis**

(dimitrios)

cheatography.com/dimitrios/

Published 21st September, 2016.

Last updated 1st January, 2017.

Page 1 of 1.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>