Loops em JS

Apenas o que for novidade.

Ainda no básico de JS, talvez hajam outros *loops* depois.

Os loops são **STATEMENTS**. Prova: passar um loop como parâmetro de função.

Não é recomendado usar *loops* para iterar sobre Arrays. Preferir os *helper methods* do Array prototype que vêm por herança.

loops citados

- `for`
- `while`
- `do ... while`
- `for ... of ...`
  - novidade!
- `for ... in ...`
  - novidade!

For loop

Nada de novo. Um for loop que imprime de 0 a 4:

```
for (let i = 0; i < 5; i++) {
  console.log(i);
}
```

Topo

While loop

Igual a C e Java

```
let i = 0;
// ^-- i já precisa existir

while (i < meuArray.length) {
    console.log(meuArray[i]);
    i++;
    // última ação por iteração no foor loop
}
```

Do ... While loop

Igual a C e Java

```
let x = 0;

do {
    console.log(meuArray[x]);
    x++;
} while (x < meuArray.length);
```

Topo

For in & For of

Novidade (eu acho)

me parece o `for each`

For In

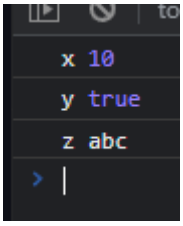
Opção recomendada para iterar sobre objetos

```
// Uso com Objetos
const meuObjeto = {
    x: 10,
    y: true,
    z: "abc"
}

for (let propriedade in meuObjeto) {

    console.log(propriedade, meuObjeto[propriedade]);
}
// --------------------
// Uso com Arrays
const meuArray = [true, 10, "abc", null];
```

```
for (let chave in meuArray) {
  console.log(meuArray[chave]);
}
```



Topo

For Of

Mais novo (ES6)

```
for (Elemento of Iterable) {
  ...
  ...
  ...
}
```

Definição de `Iterable`:

qualquer variável que possui Iterator *builtin* ou customizado. Todo Array possui um Iterable *builtin*. Exemplo de `Iterable` *values*: arrays e strings.

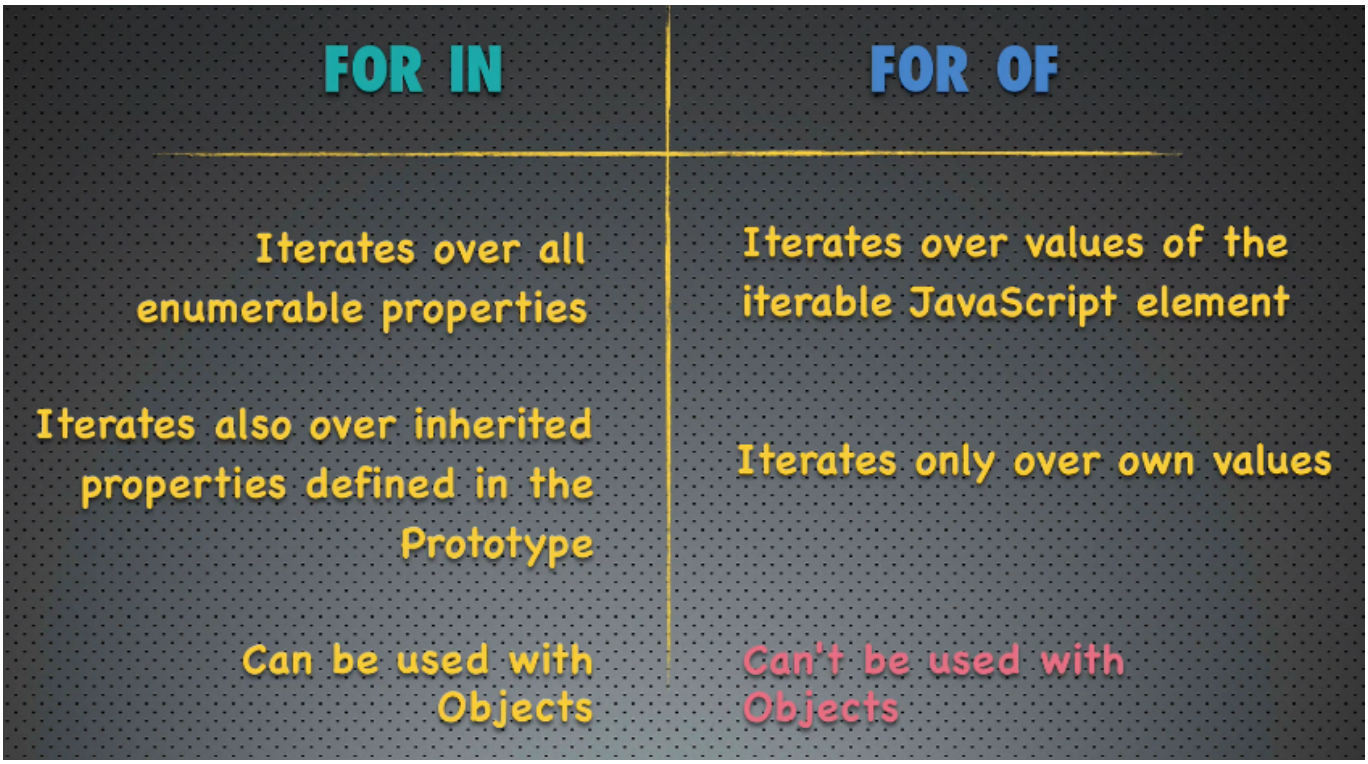
> **Não funciona** *out of the box* com objetos! É preciso definir um Iterator **customizado**!

Se não funciona com objetos, objetos não são `Iterable`.

```
const meuArray = [true, 10, "abc", null];

for (let elemento of meuArray) {
    console.log(elemento);
}
```

Diferença entre os dois



Propriedades enumeráveis: Propriedades de tipo **primitivo**. Não funciona com *métodos* (funções como propriedade).
