



# Compilado - Aula 07

- Quando implementamos algoritmos, em geral queremos economizar nosso tempo de trabalho manual. E isso é particularmente útil quando alguma coisa precisa ser feita várias vezes
- Por exemplo, digamos que você queira repetir alguma ação 100 vezes (quem sabe calcular a média de 100 alunos de uma vez?)
- Quando trazemos isso pra programação, podemos usar um recurso presente na maioria das linguagens chamado **loop**
- Um loop é uma forma de você dizer pro computador: enquanto uma determinada condição for verdadeira, repita esse código de novo e de novo e de novo
- Assim como em várias linguagens, em JS é possível criar loops de várias formas diferentes. Veremos duas delas aqui. A primeira é usando o recurso `while`:

```
while(alguma_condicao) { // esse código será repetido sem parar, até  
  que a condição acima dê falso }
```

- Repare que enquanto a condição for verdadeira, o código ficará repetindo. Isso significa que se a condição for sempre verdadeira e nunca mudar, você terá criado um loop infinito e irá travar seu programa 🤖

```
while(true) { // isso é um loop infinito, não faça isso haha }
```

- Ou seja, se quisermos controlar a quantidade de vezes que o loop acontece, precisamos que a condição de alguma forma passe a dar `false` depois da quantidade de vezes que nos interessa. Um jeito de fazer isso é usar alguma variável pra armazenar quantas vezes o loop rodou. Por exemplo:

```
let contador = 0; while(contador < 100) { // algum código que quero
repetir 100 vezes contador = contador + 1; }
```

- Repare que agora, a cada loop, somamos 1 ao contador. Assim, quando ele chegar a 100, a condição vai passar a dar falso, e o loop será encerrado.
- Repare também que estamos agora alterando a variável contador. Quando queremos alterar uma variável, não deve-se colocar o `let` na frente. Ele deve ser usado somente no momento que criamos a variável.
- Uma observação, sempre que você quiser aumentar um número em 1 unidade, tem um jeito mais prático que é usar o operador `++`:

```
contador = contador + 1 // é o mesmo que: contador++;
```

- Por exemplo, podemos loops para gerar uma string com a frase "Não devo contar mentiras" 100 vezes:

```
function umbridgeGenerator() { let contador = 0; let frase = "";
while(contador < 100) { frase = frase + "Não devo contar mentiras";
contador++; } return frase; // retornará "Não devo contar mentirasNão
devo contar mentirasNão devo contar mentiras..." (100 vezes) }
```

- Essa construção do `while` (de criar uma variável, uma condição e um incremento) é tão comum, que existe uma outra forma de criar um loop usando menos linhas. Repare os trechos abaixo:

```
let contador = 0; while(contador < 100) { // ... contador++; }
```

- Em vez de `while`, você pode fazer a mesma coisa acima usando a segunda forma de se criar loops em JS, a construção `for`, dessa forma:

```
for(let contador = 0; contador < 100; contador++) { // esse código  
  será executado 100 vezes }
```