



Compilado - Aula 08

- Até então, temos trabalhado com variáveis que guardam 1 valor cada uma, por exemplo:

```
let nota1 = 5; let nota2 = 7; let nota3 = 10;
```

- Porém, um caso muito comum na computação é quando precisamos armazenar **listas** de valores. Por exemplo, se quisermos guardar os nomes de todos os alunos, teríamos que fazer algo como:

```
let aluno1 = "Ana"; let aluno2 = "Bia"; let aluno3 = "Carlos"; ...
```

- Isso fica muito pouco prático e dificulta manipularmos essas informações. Para resolver isso, em muitas linguagens de programação existe o conceito de listas de valores, que chamamos de **arrays**. Em JavaScript podemos criar arrays usando colchetes `[]`:

```
let alunos = ["Ana", "Bia", "Carlos"];
```

- O mais interessante dessa construção é que podemos acessar os valores de um array usando a posição deles (começando em 0). No exemplo acima, Ana está na posição `0`, Bia na posição `1` e Carlos na posição `2`. Por exemplo, podemos retornar a Bia fazendo:

```
// posições 0 1 2 let alunos = ["Ana", "Bia", "Carlos"]; return  
alunos[1]; // Retornará "Bia"
```

- Podemos também obter o tamanho de uma array, usando a propriedade `.length`:

```
let alunos = ["Ana", "Bia", "Carlos"]; return alunos.length; //  
Retornará 3 pois a array tem 3 itens
```

- O mais interessante de usarmos arrays é que como podemos acessar os valores pelas posições numéricas, é um prato cheio pra usarmos loops. Lembrando que nos loops temos um "contador" que começa como 0 e vai até o número que quisermos (até o final da array, que tal?). Então se precisamos fazer alguma coisa pra cada elemento de uma array, poderíamos fazer:

```
let alunos = ["Ana", "Bia", "Carlos"]; for(let contador = 0; contador  
< alunos.length; contador++) { // esse código repetirá 3 vezes, o  
número de itens na array }
```

- Como temos essa variável contador dentro do loop, podemos usar isso pra ir acessando as posições da array. Por exemplo, pra concatenar todos os nomes:

```
let alunos = ["Ana", "Bia", "Carlos"]; let nomes = ""; for(let  
contador = 0; contador < alunos.length; contador++) { nomes = nomes +  
alunos[contador]; }
```

- Uma forma mais comum de nomear essa variável `contador` é usar a letra `i` (de índice), pra ficar mais curto e legível, mas é só um nome como outro qualquer:

```
let alunos = ["Ana", "Bia", "Carlos"]; let nomes = ""; for(let i = 0; i < alunos.length; i++) { nomes = nomes + alunos[i]; }
```

- Dando um outro exemplo, se quiséssemos fazer uma função que soma todos os valores de um array poderíamos fazer:

```
function somarTudo(lista) { let soma = 0; for(let i = 0; i < lista.length; i++) { soma = soma + lista[i] } return soma; }
```

- No código acima, se passássemos uma lista `[10, 20, 30]` por exemplo, ela retornaria a soma: `60`