Listas (Arrays)

* Array: lista
* As listas são representadas pelos **colchetes** e seus valores são separados por **vírgulas** no js

Exemplo:

const alunos = **[**‘João’, ‘Vitor’, ‘Marina’**]**;

* Para obter um item isolado da lista, utilizamos a instrução com o índice do item desejado entre colchetes da mesma forma que é feito em Python
* Para adicionar itens ao fim de uma lista, usamos a notação **nome-da-lista.push(o-que-queremos-adicionar);**
* Para adicionar um item numa **posição específica** da lista, usamos a notação **nome-da-lista[índice-que-desejamos-adicionar-algo] = algo**;
* **Obs.:** mesmo que o índice que tentarmos adicionar não existem, o item é criado naquele índice
* Para substituir um item, utilizamos a mesma notação acima: **nome-da-lista[índice-que-desejamos-substituir] = algo**;
* Podemos iniciar uma lista vazia: **lista = []**;
* Podemos ter tipos diferentes dentro da mesma lista (str, int, float etc.)
* Para **remover** **o último item** da lista, usamos a notação: **nome-da-lista.pop()**;
* Para **remover** **o primeiro item** da lista, usamos a notação: **nome-da-lista.shift()**;
* Para saber o tamanho de uma lista, usamos a notação: **nome-da-lista.length** (é usado dentro do console.log e não precisa de parênteses depois do método)

Estruturas de Repetição

* Notação do **for** no js: for (let index = 0; index < array.length; index++), onde:
* **let index = número**: define a partir de qual número começa a iteração e declaração da **variável de controle** que nesse é a index
* **index < array.length**: **estrutura condicional** que retorna um booleano, condição de continuação (caso for **true**) ou parada da iteração (caso for **false**)
* **index++**: o que acontece após a finalização de cada iteração (o **++** incrementa +1 na variável de controle
* O **for** é utilizado para percorrer uma quantidade determinada de vezes
* Para iterar sobre cada letra de uma string, fazemos o seguinte:

const nome = ‘Eduardo N Almeida’;

for (let i = 0; i < **nome.length**; i++) {

console.log(nome**[i]**);

}

* Apagar uma linha toda no vscode: **Ctrl + Shift + k**
* A **depuração** só funciona se colocarmos o **breakpoint** (ponto vermelho no ínicio da linha que aparece ao clicar) em alguma linha do código
* Podemos executar linhas de código pelo console de depuração também