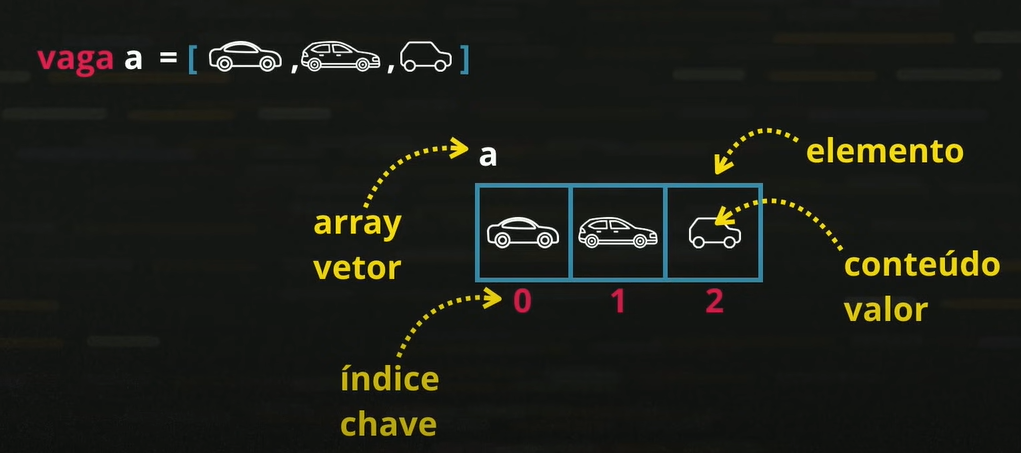
**MÓDULO 6 – VARIÁVEIS COMPOSTAS**

* Variáveis compostas devem ser capazes de armazenar vários valores em uma mesma estrutura. Já que a simples armazena apenas um valor por vez.

Para armazenar vários valores numa só variável. Usamos os vetores, que são declarados entre colchetes. Ou seja, uma variável composta é um array (vetor) – O vetor é composto de elementos. Um elemento de um vetor, é um par que agrupa o espaço da memória, o valor colocado dentro dele e o índice. O índice é o número que está embaixo, que também pode ser chamado de chave. E por último está o conteúdo de cada elemento.



* Isto é uma variável composta (array)

let num = [5, 9, 9, 2, 7];

console.log(num);

* Para adicionarmos um elemento ao array, fazemos assim: Colocamos o valor que queremos e a posição que queremos entre colchetes.

let num = [5, 9, 9, 2, 7];

num[5] = 1;

* Para colocar no final, sem sabermos qual é o valor do final do array, usamos o comando .push() que é um método interno que acrescenta valores a um array.

let num = [5, 9, 9, 2, 7];

num[5] = 1;

num.push(7);

.push e entre parêntesis o valor que você quer adicionar ao final.

* num.length serve para sabermos o número de elementos dentro de um array. Length é um atributo.

let num = [5, 9, 9, 2, 7];

console.log(`O vetor tem ${num.length} posições`);

O vetor tem 5 posições

* Para colocarmos os números em ordem crescente, usamos o num.sort().

let num = [5, 9, 9, 2, 7];

num[5] = 1;

num.push(7);

num.sort();

* Para acessarmos a posição do elemento, como já foi dito, usamos o nome da variável e colocamos a posição dentro de colchetes: Dessa forma, acessamos

let num = [5, 9, 9, 2, 7];

console.log(`O primeiro valor do vetor é ${num[1]}`);

O primeiro valor do vetor é 5

* Para escrevermos o vetor “let valores = [8, 1, 7, 5, 2, 9]” podemos escrever assim:

console.log(valores[0]);

console.log(valores[1]);

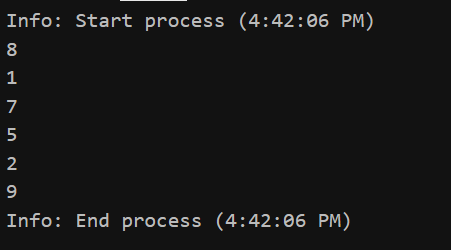
console.log(valores[2]);

console.log(valores[3]);

console.log(valores[4]);

console.log(valores[5]);

E ele imprimirá assim:



Mas, vamos pensar no trabalho que seria se tivéssemos 200 posições neste vetor.

A solução para essa situação é simples e está na aula anterior. Pois, utilizaremos a estrutura de repetição, mais especificamente o “for”.

Por quê? Vamos analisar estas linhas que fizemos:

console.log(valores[0]);

console.log(valores[1]);

console.log(valores[2]);

console.log(valores[3]);

console.log(valores[4]);

console.log(valores[5]);

A única diferença entre cada linha, são os índices que são os números de dentro do colchete que são a posição de cada elemento.

Agora imaginemos que podemos criar uma variável chamada “pos” e essa variável “pos” começar valendo 0 e for valendo ate 5 que é exatamente o limite do meu vetor.

Se eu fizer uma repetição pra isso, eu uso o console.log() apenas uma vez e coloco dentro de uma estrutura de repetição.

Criaremos uma estrutura chamada “for” que já vimos. Esse “for” tem 3 elementos – 1° Inicialização, 2° Teste lógico e 3° Incremento. E dentro do for, colocamos o console.log().

Começamos colocando a variável pos na inicialização – let pos = 0. Ou seja, a variável esta iniciando como 0 – O meu teste lógico é fazer a variável pos chegar ate 4. So que, chegar até 4 é no nosso caso, num vetor de 4 posições. Mas, e se o vetor tiver 50 posições e se for mais? E se eu não souber quantas posições tem? Tem como descobrirmos quantas posições tem, basta utilizarmos o length. Sendo assim, nosso teste lógico será: “ pos < num.length “ – Isto é, enquanto o “pos” for menor do que num.length, ou seja, enquanto ele não chegar no final do meu vetor, eu vou fazer o “ pos++ “ que é o incremento, para ele adicionar 1 cada vez que ele fizer um looping.

E dentro do console.log() colocaremos valores[pos] que ao invés de ser valores[1] ou qualquer número, o índice é a variável que inicia com 0 e vai até o fim do vetor.

for (let pos = 0; pos < valores.length; pos++) {

  console.log(`A posição ${pos} tem o valor ${valores[pos]}`);

}

Texto

Descrição gerada automaticamente

Agora, uma forma mais simples de fazermos o que fizemos agora:

Criaremos um “for” que agora não tem 3 elementos mais. A primeira coisa que colocaremos dentro desse for é o meu índice. Onde eu coloquei la em cima “ let pos = 0 “ Eu so vou colocar “ let pos “ e do lado colocarei um “in” e depois do in, basta colocar o nome da variável composta, que no nosso caso se chama “valores”.

Isso é, para cada posição dentro de valores (é assim que leremos) o “for” a gente lê “para” o “in” a gente lê “dentro”. Então esse código:

for (let pos in valores) {

  console.log(`A posição ${pos} tem o valor ${valores[pos]}`);

}

A gente lê assim: Para(let) cada posição em/dentro (in) variável composta(valores), eu vou mostrar valores[pos].

* Por fim, veremos apenas como buscar valores dentro de um vetor.

valores. indexOf() é um método que busca valores dentro do vetor. Dentro dos parêntesis, eu coloco o VALOR que eu quero procurar. O javaScript achando o valor dentro do vetor, ele te retornará a posição que esse valor se encontra.

Se pesquisarmos um valor e o JavaScript não encontrar, ele retornará -1.

let num = [5, 9, 9, 2, 7];

console.log(`O primeiro valor do vetor é ${num[0]}`);

console.log(`O vetor tem ${num.length} posições`);

let pos = num.indexOf(9);

console.log(pos);

Texto

Descrição gerada automaticamente