

Array é uma lista, é uma coleção de coisas

Você aprendeu que a gente tem string onde a gente pode colocar um texto mas se você precisar colocar numa variável uma lista de alunos:

```
const alunos = 'Luiz Maria João';
```

Eu não tenho divisões, não posso por exemplo ver o nome do Maria

Como criar um array:

```
const alunos = ['Maria', 'Luiz', 'João']; // Lista de nomes
```

Eu vou tentar um tipo de dado dentro do array, mas eu posso colocar outros valores como booleano, funções etc o JS permite mas não é uma boa prática de programação!

Então só use 1 tipo de dado.

```
console.log(alunos);
```

As strings são indexadas elas tem os índices: 0123...

Os Arrays também são indexados mas são diferentes das strings porque os arrays indexados por elementos. Então o Luiz é o índice 0 etc

Eles não indexados pelo seu valor, e sim por elemento.

Pra acessar a letra de uma string seria:

```
console.log(nome[10]);
```

Nos arrays isso funciona da mesma maneira:

```
console.log(alunos[0]);
```

Cada um dos índices representa 1 elemento e todos os elementos representam todos os arrays.

Com array eu posso editar um valor:

```
alunos[0] = 'Eduardo';  
console.log(alunos);
```

Eu posso adicionar um valor:

```
alunos[3] = 'Luiza'; // Esse índice não existe então ele é adicionado
```

Dessa maneira pode ser complicado, eu teria que saber qual é o tamanho dele (quantos elementos ele tem), qual é o último índice dele e daí falar que eu quero adicionar um elemento do último índice que neste caso é o 3.

Pra eu saber o tamanho do array é fazer da mesma forma na string:

```
console.log(alunos.length); // 3 é depois do último elemento
```

Eu posso fazer:

```
alunos[alunos.length] = 'Luiza';  
alunos[alunos.length] = 'Leonardo';  
alunos[alunos.length] = 'Luan';  
console.log(alunos);
```

Ele vai adicionando esses nomes ao array, o array vai aumentar então ele tem 6 elementos.

```
[ 'Maria', 'Luiz', 'João', 'Luiza', 'Leonardo', 'Luan' ]
```

Mas tem uma função que faz isso para adicionar depois do último elemento.

```
alunos.push('Leonardo');
```

E se eu quiser adicionar ao começo do array?

Bom, eu não posso ter um índice -1 então eu preciso mover o Luiz para o índice 1 a Maria para o 2 e assim por diante. Porém nós temos uma função que faz isso:

```
alunos.unshift('Luiza');  
alunos.unshift('Léo');
```

Primeiro eu adicionei a Luiza no 1° índice, ela já estava no 1°
E aí eu pedi para adicionar o Léo no 1° então ela foi para o 2° índice;

```
[ 'Léo', 'Luiza', 'Maria', 'Luiz', 'João' ]
```

Funções para usar nos Arrays:

pop -> Remove elementos do final, o último elemento sairá

```
alunos.pop();
```

```
[ 'Maria', 'Luiz' ]
```

O João saiu.

A gente pode salvar o elemento que nós estamos removendo do array em uma variável:

```
const removido = alunos.pop();  
console.log(removido);  
console.log(alunos);
```

João

```
[ 'Maria', 'Luiz' ]
```

Similar ao pop temos o:

shift -> Remove o primeiro elemento

```
const removido = alunos.shift();  
console.log(removido);  
console.log(alunos);
```

Tudo isso altera os índices. Eu quero apagar o índice 1 (Maria)

```
delete alunos[1];
```

Como os meus índices não serão alterados eu vou ficar com o elemento vazio:

```
[ 'Maria', <1 empty item>, 'João' ]
```

O índice 1 ficou vazio. Os índices não foram alterados mas ele está vazio.

Se eu quiser acessar um índice que não existe ele retorna Undefined:

```
console.log(alunos[50]);
```

Ele só vai dar erro quando você tentar acessar alguma coisa dentro do Undefined mas isso vamos ver mais pra frente.

```
const alunos = ['Maria', 'Luiz', 'João']; // Luiza, Eduardo
```

```
alunos.push('Luiza');
```

```
alunos.push('Eduardo');  
console.log(alunos);
```

Suponha que eu queria fatiar esse array, é muito similar com que a gente fez nas strings

por exemplo eu quero pegar o elemento 0 até o 2:

```
alunos.slice(0, 3);
```

Eu coloquei 0 até o 3 porque se eu coloco 0 até o 2 ele não será incluído então eu coloco até o índice 3 para o índice 2 ser exibido.

```
[ 'Maria', 'Luiz', 'João' ]
```

Eu posso colocar valores negativos para fatiar também.

Ele vai usar o tamanho do array - o número negativo

Meu array tem 5 elementos eu quero -1 então ele vai parar na Luiza

```
alunos.slice(0, -1); // 5 - 1 = 4
```

```
[ 'Maria', 'Luiz', 'João', 'Luiza' ]
```

É a mesma coisa que a gente nas strings, só que agora os índices dos arrays não são por letras, por caracteres eles são por elementos separados por ,

Array no JavaScript é considerado um objeto porque um array é realmente um objeto indexado

é muito similar a única diferença é que o array é indexado automaticamente e no objeto é um pouco diferente.

Mas se você quiser ter certeza de que você está trabalhando com um array você faz:

```
console.log(alunos instanceof Array);
```

Eu estou perguntando se alunos é uma instância de array, ele retorna true or false.

Eu posso alterar o meu array também, o tipo dele:

```
let alunos = ['Maria', 'Luiz', 'João'];  
alunos = 123;
```

Nesse caso ele não é um Array então retorna false;

Geralmente, na maioria dos casos você vai adicionar elementos no final porque dependendo da quantidade de elementos que tem no Array, você quer mudar os índices de todos os elementos no Array pode ser ruim para a performance do programa porque ele tem que mexer em todos elementos do Array então é geralmente melhor adicionar ao final do Array, a gente adiciona ou remove elementos do final porque assim a gente não afeta os índices dos outros elementos no Array.