Aprenda JavaScript moderno (ES6+) para front-end e back-end

Este documento inclui aprendizado sobre JavaScript e TypeScript, você vai aprender ambas as linguagens no mesmo pacote.

Inicialmente, você aprenderá a utilizar recursos nativos do JavaScript sem a necessidade de utilizar frameworks ou bibliotecas adicionais.Trabalharemos tanto no Node.Js (back-end) quanto no navegador (front-end).

1. JavaScript Funçoes

Declaração de funções

- Function hoisting

O motor do Javascript eleva as declarações de funções e variáveis para o topo do JavaScript, isso significa que você pode fazer a chamada da função tanto antes ou depois da declaração da função.

Ex: chamada pós declaração.

```
// Declaração de função (Function hoisting)
function falaOi(){
    console.log('Oie');
};
falaOi();
```

Ex: Chamada antes da declaração

```
// Declaração de função (Function hoisting)
falaOi();
function falaOi(){
    console.log('Oie');
};
```

❖ Function expression

A função pode ser tratada como dado, ou seja, você criar uma função e tratá-la como dado.

Ex: Tratando função como dado e passando a função como parâmetro.

```
// First-class objects (Objetos de primeira classe)

// Function expression
const souUmDado = function(){
    console.log('Sou um dado.');
};
souUmDado();
function executaFuncao(função){
    console.log('Vou executar sua função abaixo:');
    função();
}
executaFuncao(souUmDado);
```

❖ Arrow function

Trata-se um recurso novo do **ES6** é uma declaração de função mais curta, porém, o resultado é o mesmo.

Ex: Arrow function

```
// Arrow function
const funcaoArow = () => {
    console.log('Sou uma arrow function.');
}
funcaoArow();
```

❖ Function dentro de um objeto

A função dentro de um objeto é tratada como se ela fosse um método do objeto.

Ex:

```
// Dentro de um objeto
const obj = {
    falar(){
        console.log('Estou falando ...')
    }
}
obj.falar();
```