JavaScript Intermediário

- Template String
  - Material
  - Objetivo
  - VAR
  - LET
  - Template String
- Operador Ternário e Arrow Functions
  - Material Complementar
  - Objetivo
  - Const
  - Exemplos
  - Arrow function | `const Pedro = () => {  }`
  - Exemplo
  - Construção dinâmica de objeto
- O operador Spread
  - Objetivo
  - Exemplo
  - Exemplo 2
- Desestructuring e Match
  - O que é?
  - Exemplo sem Destructuring
  - Exemplo com Desctructruing
  - Match
  - RegEXP
  - Exemplo
- SPA, PWA e WebComponents
  - Single Page Application
  - Progressive Web Apps
  - Web Componentes
  - Componentes Player

Template String

Material

- Uma reintrodução ao JS

Objetivo

- Atalho e dicas para evoluir a sua perfomance como desenvolvedor

VAR

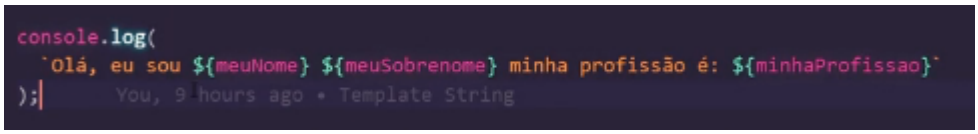
- Coloca muita coisa em memória e tem escopo global e pode dar problema

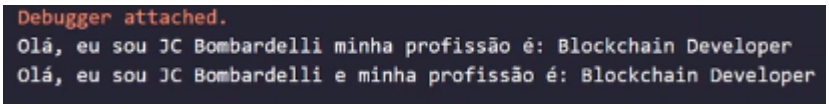
LET

- Vamos usar muito LET nos nossos códigos

Template String

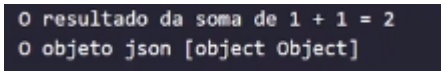
- Forma de escrever Strings com variáveis JavaScripts dentro
- Usaremos o caractere `` `${ }` ``, com uma variável dentro.
- Fizemos a interpolação de texto com variável em JS





- Podemos ainda executar operações dentro das template strings

- Podemos ainda usar para um objeto em **JSON**



Operador Ternário e Arrow Functions

Material Complementar

- developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/

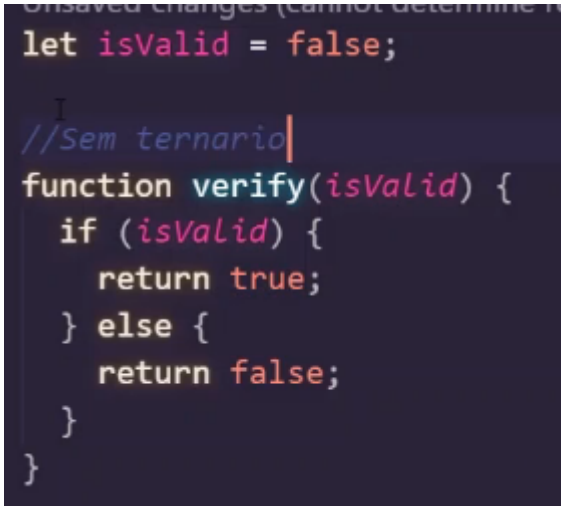
Objetivo

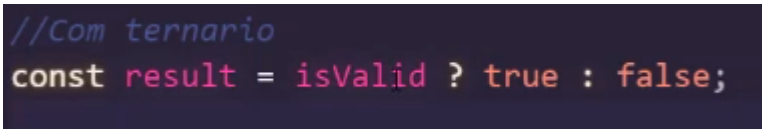
- Entender a evolução e como transformar simples decisões em um ternário, com mais produtividade e legibilidade.
- É um Syntax Sugar.
- Substitui a estrutura `If...else`

Const

- O valor atribuído não pode ser mais alerado
- Não vai ser permitir ser alterada sem querer

Exemplos





Arrow function | `const Pedro = () => {  }`

- É uma abstração de uma função

Exemplo

- Com e sem Arrow Function

```
function soma(x, y) {
  return x + y;
}

const multiplicacao = function(x, y) {
  return x * y;
};
```

```
const dividir = (x, y) => {
  return x / y;
};
```

Construção dinâmica de objeto

```
const objeto = () => ({ nome: 'JC', sobrenome: 'Bombardelli' });
```

O operador Spread

Objetivo

- Administrar os dados que estão sendo passados no JavaScript
- Pegar o valor de uma lista e coloca onde queremos.

Exemplo

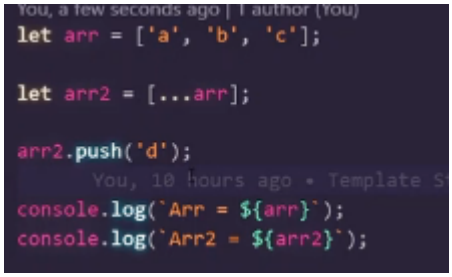
- Insirar elementos em um array pre-existente.
- Para cada elemento da lista incluir vai percorrer o elemento da lista principal

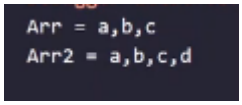
```
console.log(lista);      You, 9 hours ago • Templat

const listResult = [1, 2, ...listaIncluir, 5];
```

Exemplo 2

- Queremos criar uma cópia



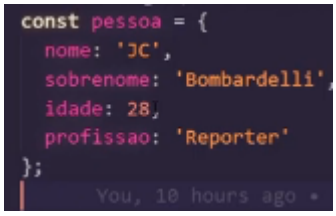


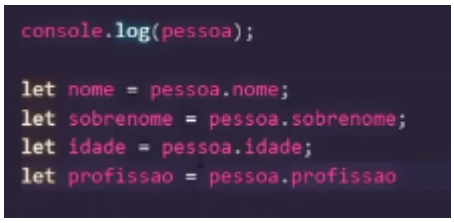
Desestructuring e Match

O que é?

- Usados em objeto para alterarmos chaves e valores dentro deles.
- Serve para processar e quebrar informações e aproveitar apenas as propriedades que quisermos

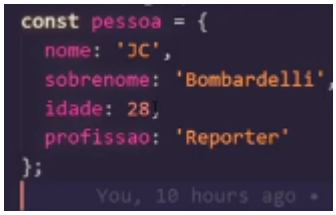
Exemplo sem Destructuring

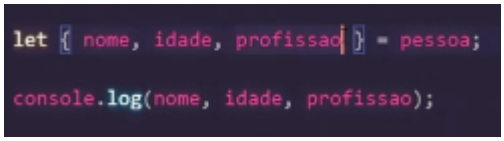


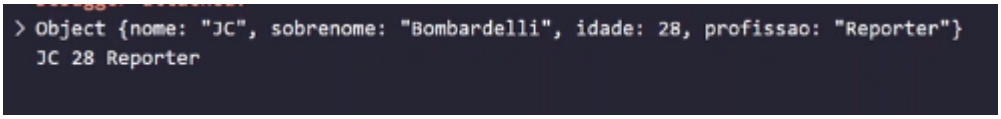


Exemplo com Desctructruing

- Passamos apenas as propriedades que nos interessa entre `chaves`
- Caso não ache a propriedade, mostrará undefined







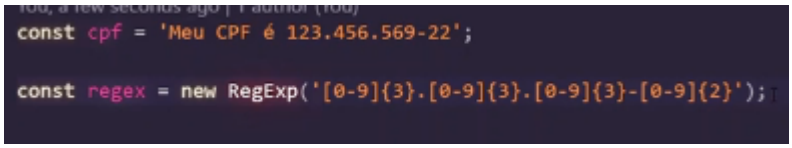
Match

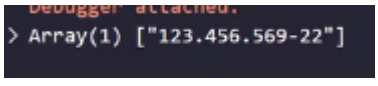
- Buscar conteúdos específicos sobre uma massa de dados que não temos certeza se informação está lá ou como está estruturado lá dentro

RegEXP

- É uma expressão regular que permite fazermos máscaras e selecionarmos determinada combinação de carácteres.

Exemplo





SPA, PWA e WebComponents

Single Page Application

- Nos últimos anos temos usado muito SPA para construção dos nossos sites através de Frameworks, que dão ganho de perfomance

    - React (fb) Angular (google), VueJS (Single Components)

Progressive Web Apps

- Seu software rodando em diversos lugares

Progressivo

- Funcionando para qualquer usuário em qualquer navegador escolhido

Responsivo

- Adequando-se a qualquer formato, seja ele desktop, celular, tablet etc.

Independente de conectividade

- Aprimorado através de Service Workers para trabalhar offline ou com conexão limitada.

Semlhante a Aplicativos

- Deve ter interface similar a um app para usuários, oferecendo os mesmos recursos independentes do dispositivos

Sincronizado

- Mantendo-se atualizado constantemente pelo Service Workers
    - Funcionar offline e sincronizará quando online

Seguro

- Sempre ocm conexão (quando disponível) via HTTPS

Reenvolvente

- Deve gerar engajamento através de notificações PUSH

Instalável

- Possibilidade de guardar o aplicativo sem necessariamente instalá-lo

Linkável

- Facilmente compartilhável com um link, reduzindo o atrito das lojas de aplicativos.

Web Componentes

- Web nua e crua como frameworks

HTML Template

- Possibilita a criação de fragmentos HTML que podem ser invocados sempre que necessários
    - EJS
    - Temos ideia de componentes

Custom Elements

- Capacidade de criar componentes customizados.

Shadow DOM

- Estilos e diretivas globais ficam encapsulados, ou seja, cada componente não é capaz de interferir nos demais.
- Funcionam como módulos.

ES Modules

- Possibilidade de carregar módulos nativamente

Componentes Player

- Alguns compiladores.