TypeScript

Created	@March 29, 2022 10:28 PM
Created by	
Tags	Front End

O que é o Typescript:

- É JavaScript com sintaxe de tipos
- É uma linguagem de Programação
- É um superset de Javascript, ou seja, é uma linguagem construída “em cima” do Javascript (recursos extras)
- Todo código Javascript válido é código Typescript válido
- A relação entre Typescript e Javascript pode ser vista como semelhante a do SCSS e o CSS
- É uma linguagem fortemente tipada/ estática (não consegue modificar)

Por que e quando utilizar o Typescript:

- Aumenta a produtividade
- Permite que a sua IDE (VSCode) seja “mais inteligente”, por exemplo, apontando erros e mostrando opções de autocomplete
- Permite que você identifique erros em tempo de compilação ao invés de em tempo de execução
- Curva de aprendizado reduzida, visto que ele usa a sintaxe do JS e é possível adotar o TS de forma parcial e gradativa

- Traz recursos extras que não existem nativamente no JS, como tuplas, enums, melhor suporte a POO, etc
- Vantajoso em projetos grandes
- Importante ter bom conhecimento em JS
- Mesmo em um projeto escrito com TS é importante saber quando usar suas funcionalidades

Principais recursos do Typescript:

- Tipagem para variáveis, objetos, parâmetros e retornos de funções
- Criação de nossos próprios tipos e interfaces
- Checagem de erros pela IDE enquanto escrevemos o código
- Função de autocompletar da IDE
- Excelente documentação e suporte da comunidade

O que é tipagem e como funciona no Typescript:

- **Tipos Primitivos:**
  - Mais básicos e mais utilizados (typeof - JS)
  - Existem três mais utilizados:
    - Boolean (true ou false) → `let example: boolean = true`
    - Number (número → não diferencia inteiros e pontos flutuantes) → `let example: number = 2`
    - String (texto) → `let example: string = “hello”`
  - Também o Array, que representa listas de dados
    - A sintaxe básica para especificar um array é utilizando o tipo dos seus elementos. Por exemplo, *string[]* ou *number[]* → `let example: number[] = [1,2,3,4]` ou `let example : Array<number> = [1,2,3,4]`

        - Usamos arrays como sendo uma lista onde todos os elementos tem o *mesmo tipo*, mas esse comportamento também pode ser evitado

- **Inferência de tipo:**
    - : void → retorno da função
    - Em TypeScript, existem vários locais onde a inferência de tipos é usada para prover informação quando não se tem um tipo explícito de anotação.
    - Se colocar o mouse em cima de uma variável, aparece a tipagem → VSCode faz automaticamente → Sempre será esse tipo
    - Com o mouse em cima da função, clicar em Quick Fix → if all types (vai completar os tipos automaticamente)

7 - Inferência de Tipos

https://onebitcode.notion.site/7-Infer-ncia-de-Tipos-60849cb6a0f3422bb80bf7c190927ee9

Como instalar, configurar e usar o Typescript na prática:

Iniciar:

No terminal → pasta do projeto → comandos:

npm init → Iniciar o projeto com node

npm install -g typescript → instala globalmente na máquina → pode ser usado em qualquer projeto

npm install —save-dev typescript → quando não precisar em produção → por projeto

Compilar:

comandos:

npx t*sc* ***nomedoarquivo*** → vai criar um arquivo ***nomedoarquivo****.js* → sempre usar quando atualizar (pode utilizar no node)

- em arquivo package.json → “Scripts”:{ “build” : “*tsc* ***nomedoarquivo****.ts*”}

  excluir o ***nomedoarquivo****.js*

no terminal → *npm run build* (atualizara e compilara automaticamente)

8 - Como Instalar e Usar o Typescript

https://onebitcode.notion.site/8-Como-Instalar-e-Usar-o-Typescript-3163f3a6f9ae495f8764696d8d1367ee

Como criar os seus próprios tipos:

Programação orientada a Objetos com Typescript: