



# Introdução a TypeScript



**Tecnólogo em Análise e Desenvolvimento de  
Sistemas - Faculdades SENAC/PE**

**Professor:** Danilo Farias

- Conhecer os motivos pelos quais o TypeScript é um aprimoramento em relação ao JavaScript para desenvolvimento para a Web.
- Selecionar um editor de TypeScript.
- Instalar o TypeScript.
- Configurar um projeto do TypeScript no Visual Studio Code.

# Antes de falar sobre TypeScript, vamos conversar sobre JavaScript



- O **JavaScript**, uma das linguagens de programação mais usadas do mundo, tornou-se a linguagem de programação oficial da Web. Os desenvolvedores o usam para escrever aplicativos multiplataforma que podem ser executados em qualquer plataforma e em qualquer navegador.
- Atualmente, o maior mantenedor da linguagem é a **Fundação Mozilla**;
- Encontramos ótimos materiais e tutoriais sobre JavaScript na W3School, mas também encontramos referência completa do JavaScript no site do Mozilla:
  - <https://developer.mozilla.org/en-US/docs/web/javascript>
  - <https://www.programiz.com/javascript/get-started>
  - <https://www.w3schools.com/js/default.asp>

- Embora o **JavaScript** seja usado para criar aplicativos multiplataforma, ele não foi concebido para aplicativos grandes envolvendo milhares ou até mesmo milhões de linhas de código.
- O **JavaScript não tem alguns dos recursos de linguagens mais maduras** presentes nos aplicativos sofisticados de hoje.
- O **TypeScript** resolve as limitações do JavaScript, fazendo isso sem comprometer a principal proposta de valor do JavaScript:
  - A capacidade de executar seu código em qualquer lugar e em todas as plataformas, navegadores e hosts.

# O que é TypeScript



- O TypeScript é uma **linguagem de software livre** desenvolvida pela **Microsoft**. Ele é um **superconjunto do JavaScript**, o que significa que você pode usar as habilidades de JavaScript que já desenvolveu junto com determinados recursos que não estavam disponíveis para você anteriormente.
- O principal recurso do TypeScript é o **sistema de tipos**. No TypeScript, você pode **identificar o tipo de dados** de uma **variável** ou **parâmetro** usando uma dica de tipo. Com as dicas de tipo, você descreve a forma de um objeto, o que proporciona uma **melhor documentação** e permite que o TypeScript valide se o código está funcionando corretamente.

The TypeScript logo, consisting of a blue square with the white letters "TS" inside.

TS

# Vamos experimentar TS?



- Abra o [Playground](#) do TypeScript. O Playground é uma plataforma de compilação rápida para ensino de TypeScript.
- Exemplo de código JavaScript no painel do TypeScript:

```
function addNumbers(x, y) {  
    return x + y;  
}  
  
console.log(addNumbers(3, 6));
```

# Vamos experimentar TS?



- Ao executar o código JavaScript.
- Em seguida, selecione a guia Logs e observe que o valor 9 é registrado no console.
- O JavaScript atribuiu o tipo **number** aos parâmetros x e y e a função retornou um número.

# Vamos experimentar TS?



- Substitua 3 por "3" (incluindo as aspas) no código do TypeScript e execute-o.
- O JavaScript agora atribui o **string** tipo ao parâmetro x e retorna "36", um tipo **string**, para o console.
- Pelo fato do JavaScript ser fracamente tipado ele vai variar os tipos dos valores e tentar resolver a operação de soma, concatenando a string "3", a nova string "6";



# Vamos experimentar TS?



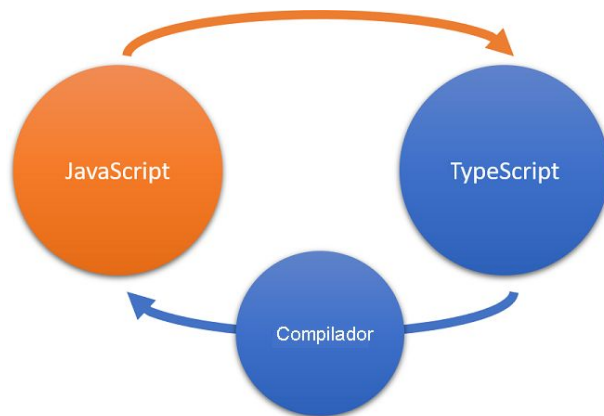
- Atualize o código TypeScript para especificar um tipo para cada parâmetro.
  - **Substitua x por x: number e y por y: number.**
  - Você observará que os erros agora desapareceram dos parâmetros, mas um novo erro apareceu no primeiro argumento na chamada de função:
    - "Um argumento de tipo 'string' não pode ser atribuído a um parâmetro de tipo 'number'."

- O **TypeScript** tem mais recursos de codificação que você **não encontrará em JavaScript**:
  - Interfaces
  - Namespaces
  - Genéricos
  - Classes abstratas
  - Modificadores de dados
  - Opcionais
  - Sobrecarga de função
  - Decoradores
  - Utilitários de tipo
  - Palavra-chave readonly

# Compatibilidade do TypeScript com o JavaScript



- O TypeScript é um superconjunto estrito do [ECMAScript 2015](#) (ECMAScript 6 ou ES6). Essa relação significa que todo o código JavaScript também é código TypeScript, e um programa TypeScript pode consumir o JavaScript de maneira direta.



- O Visual Studio Code é um editor de código-fonte leve e gratuito executado em sua área de trabalho nos sistemas operacionais macOS, Linux e Windows.
- Ele tem suporte interno para JavaScript, TypeScript e Node.js, além de um ecossistema avançado de extensões para outras linguagens e runtimes.



- O Visual Studio Code é um editor de código-fonte leve e gratuito executado em sua área de trabalho nos sistemas operacionais macOS, Linux e Windows.
- Ele tem suporte interno para JavaScript, TypeScript e Node.js, além de um ecossistema avançado de extensões para outras linguagens e runtimes.



- O **Node.js**, uma biblioteca do JavaScript para aplicativos do lado do servidor, não é necessário para aprender o TypeScript. Mas, ao instalá-lo, você também obtém o Gerenciador de Pacotes do Node, também chamado de **npm**, o gerenciador de pacotes da linguagem JavaScript. Em seguida, você usará o npm para instalar o pacote TypeScript.
  - Vá para a página de [Downloads do Node.js](#).



- O TypeScript está disponível como um pacote no registro npm como typescript.
- Para instalar a versão mais recente do TypeScript:
  - Na janela do Prompt de Comando, insira
    - **npm init -y**
    - **npm install -g typescript**
    - **npm i -D tsx**
  - Insira **tsc** para verificar se o TypeScript está instalado. Se ele tiver sido instalado com êxito, esse comando mostrará uma lista de comandos e opções do compilador.

- Nem todo código **TypeScript** é código **JavaScript**.
- O TypeScript adiciona uma nova sintaxe ao JavaScript, o que torna o JavaScript mais fácil de ler e implementa alguns recursos, como a digitação estática.
- Embora o código TypeScript torne o desenvolvimento mais fácil e menos propenso a erros, os navegadores e a maioria dos outros runtimes não dão suporte nativo a TypeScript. Por esse motivo, o TypeScript requer uma etapa de build ([transcompilador](#)) para transformá-lo em JavaScript para que o aplicativo funcione.



- Teremos que criar um projeto TypeScript com o NodeJS para isso:
  - **npx tsc --init**
  - Crie na raiz do projeto a pasta **src**
  - No tsconfig atualizar:
    - No inicio inserir --> "include": ["../src"]
    - No "compilerOptions" descomentar a linha --> "outDir":  
"./build"
  - **npx gitignore node**
  - **npm i -D ts-node-dev**

- No projeto TypeScript com o NodeJS, por fim vamos fazer esse ultimo passo:
  - Ajustar o package.json em "scripts" para:

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "dev": "ts-node-dev src/index.ts",  
  "start": "node build/index.js",  
  "build": "rimraf .build && tsc"  
}
```

- Agora vamos criar um arquivo **index.ts** na raiz do **src**
- Vamos escrever um “Hello Word” com TypeScript
  - No arquivo index.ts, escreva: `console.log("Olá Mundo!");`
- Por fim, no terminal execute o código TypeScript para sua compilação e execução usando o comando:
  - Modo de Desenvolvimento: **npm run dev**
  - Build do Projeto: **npm run build**
  - Executar o Build: **npm start**

# Dúvidas?





# Introdução a TypeScript



**Tecnólogo em Análise e Desenvolvimento de  
Sistemas - Faculdades SENAC/PE**

**Professor:** Danilo Farias