

TypeScript

Prof. Anderson Lemos

Roteiro

- Introdução ao TypeScript
- Relação com o JavaScript
- Características do TypeScript
- Instalação do TypeScript

Referência e documentação

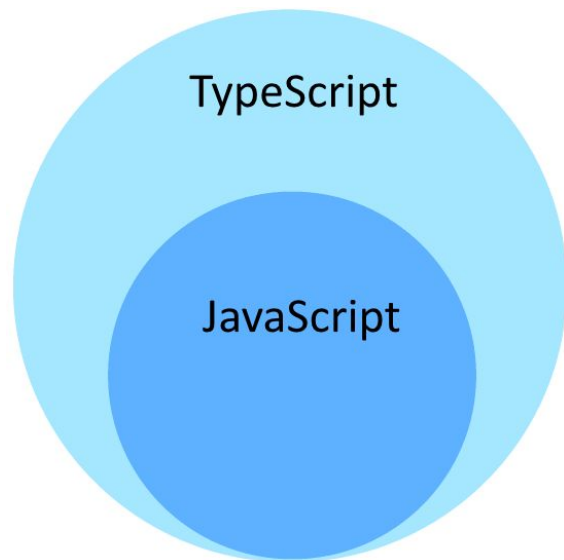
<http://www.typescriptlang.org/>

Introdução

- Linguagem de programação
 - Criada e mantida pela Microsoft
- Open Source
- Disponível para diversos sistemas operacionais
 - Linux
 - Windows
 - MAC OS
- Projetado para ser utilizado tanto no Back-end quanto no Front-end
- Relação estreita com o JavaScript

TypeScript e o JavaScript

- Superconjunto do JavaScript
 - Todo programa JS é um programa TS
 - O contrário não necessariamente é verdade (nem todo programa TS é um programa JS)
- Utilizado no Angular 2
- TypeScript é “compilado”
 - O que se faz é uma tradução para o JavaScript
 - É executado como um programa JavaScript normalmente



JavaScript é JAVA?

JAVA *is to* JAVASCRIPT *as* HAM *is to* HAMSTER



Características

- Herda as características do ECMAScript 6/2015
 - Classes
 - Módulos
 - Funções arrow
 - Parâmetros opcionais
 - Parâmetros com valor padrão
- Novas características
 - Tipagem opcional
 - Checagem de tipo em tempo de compilação
 - Inferência de tipo
 - Orientação a objetos
 - Modificadores de acesso, classes e métodos abstratos, interfaces...

TypeScript Playground

- Para testes rápidos

<http://www.typescriptlang.org/play/>

Instalação

- IDE's
 - Existem várias IDE's e programas que podem ser utilizados para desenvolver programas TS
 - Atom
 - Eclipse
 - Emacs
 - WebStorm
 - Vim
 - **Visual Studio Code**
 - **Sublime Text**

Instalação - Visual Studio Code

- Windows
 - Baixar o executável em <https://code.visualstudio.com/download>
 - Instalar com dois cliques e seguindo as instruções da tela (basicamente clicando em *next*)
- Linux
 - Baixar o arquivo .deb em <https://code.visualstudio.com/download>
 - Executar os seguintes comandos no terminal (precisa de permissão sudo):
 - `sudo apt install gdebi -y`
 - `sudo gdebi arquivo_baixado_vscode.deb`

Instalação - Sublime Text

- Windows
 - Baixar o executável em <https://www.sublimetext.com/3>
 - Instalar com dois cliques e seguindo as instruções da tela (basicamente clicando em *next*)
- Linux
 - Executar os seguintes comandos no terminal (precisa de permissão sudo):
 - `wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | sudo apt-key add -`
 - `echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/sources.list.d/sublime-text.list`
 - `sudo apt-get update`
 - `sudo apt-get install sublime-text`

Instalação - Sublime Text

- Depois de instalado o sublime, é recomendada a instalação do pacote para TypeScript
 - Instalar o gerenciador de pacotes do Sublime
 - Basta clicar em Tools → Install Package Control...
 - Instalar o pacote para TypeScript
 - Clicar em Preferences → Package Control → Install Package → TypeScript

Instalação - TypeScript

- Sublime - apenas para digitar o código
- Visual Studio Code - para digitar o código e executar, mas necessita a instalação do TypeScript
- Necessário instalar o TypeScript para compilar e executar
- O TypeScript pode ser executado como um módulo Node.js
 - Instalar o Node.js
 - Instalar o npm, o gerenciador de pacotes do Node.js
 - Instalar o módulo do TypeScript para o Node.js

Instalação - TypeScript

- Instalação no Windows




- Baixar o instalador do Node.js em <https://nodejs.org/en/download/>

Seguro | <https://nodejs.org/en/download/>

Downloads

Latest LTS Version: 8.9.4 (includes npm 5.6.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

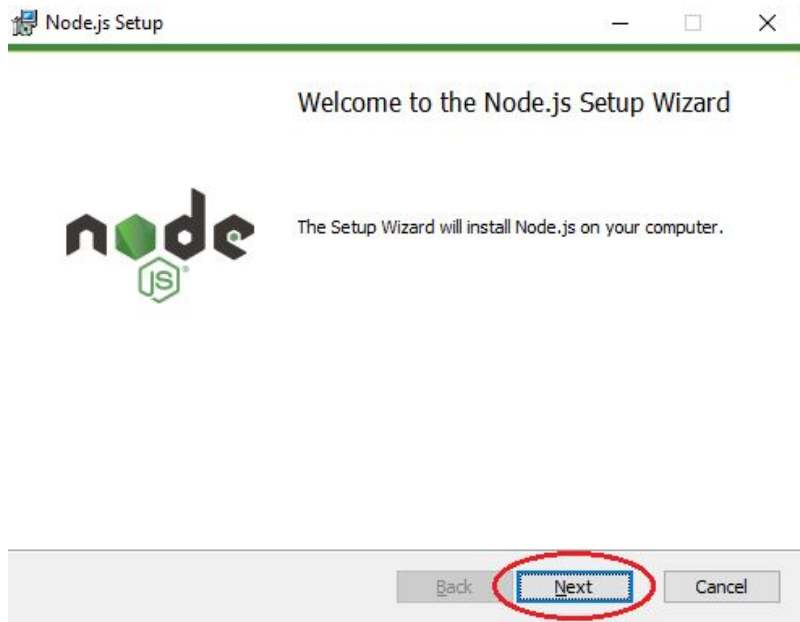
LTS Recommended For Most Users	Current Latest Features	
 Windows Installer <small>node-v8.9.4-x64.msi</small>	 macOS Installer <small>node-v8.9.4.pkg</small>	 Source Code <small>node-v8.9.4.tar.gz</small>

Windows Installer (.msi)
Windows Binary (.zip)
macOS Installer (.pkg)
macOS Binary (.tar.gz)
Linux Binaries (x86/x64)
Linux Binaries (ARM)
Source Code

32-bit	64-bit	
32-bit	64-bit	
64-bit		
64-bit		
32-bit	64-bit	
ARMv6	ARMv7	ARMv8
node-v8.9.4.tar.gz		

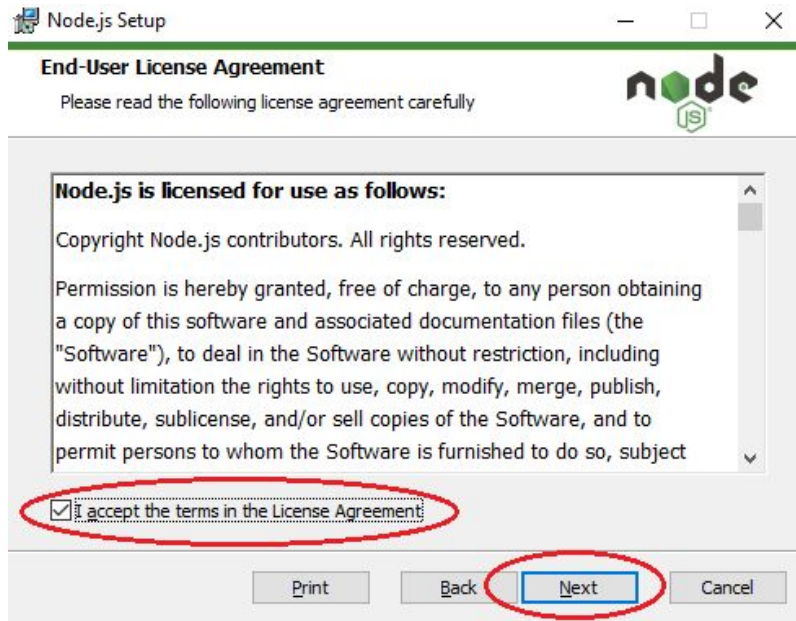
Instalação - TypeScript

- Instalação no Windows
 - Instalar com dois cliques, seguindo as instruções (basicamente clicando em *next*)



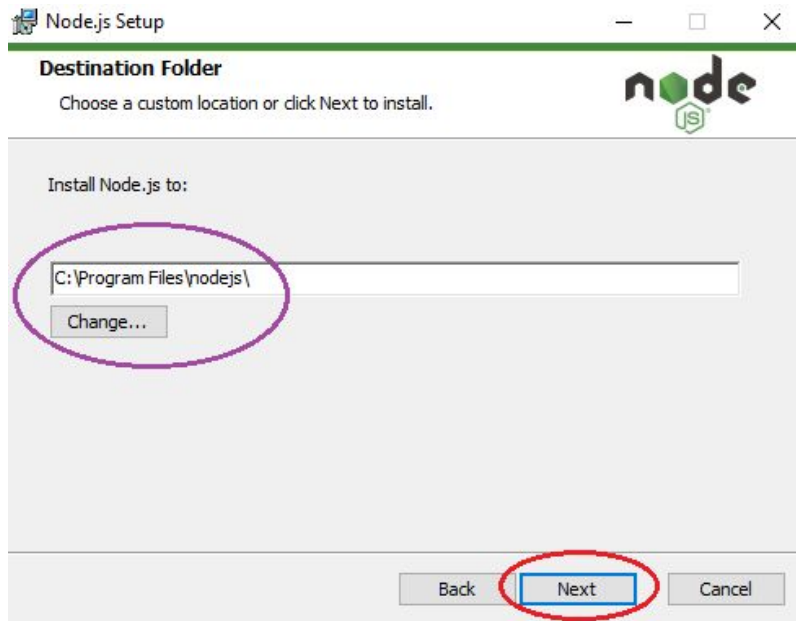
Instalação - TypeScript

- Instalação no Windows
 - Clique no checkbox para aceitar os termos de uso



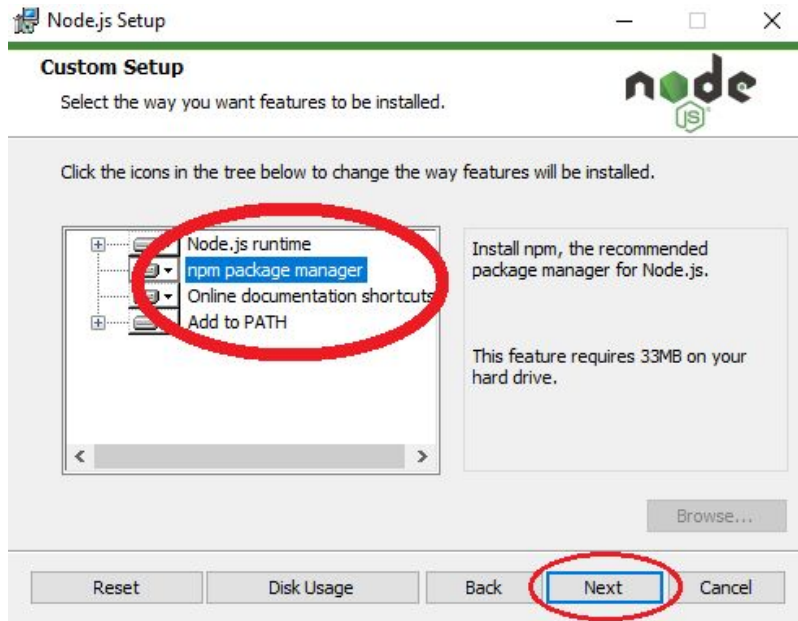
Instalação - TypeScript

- Instalação no Windows
 - Escolha a pasta onde deseja instalar o Node.js
 - Recomenda-se utilizar a pasta padrão (não modificar, apenas clicar em *next*)



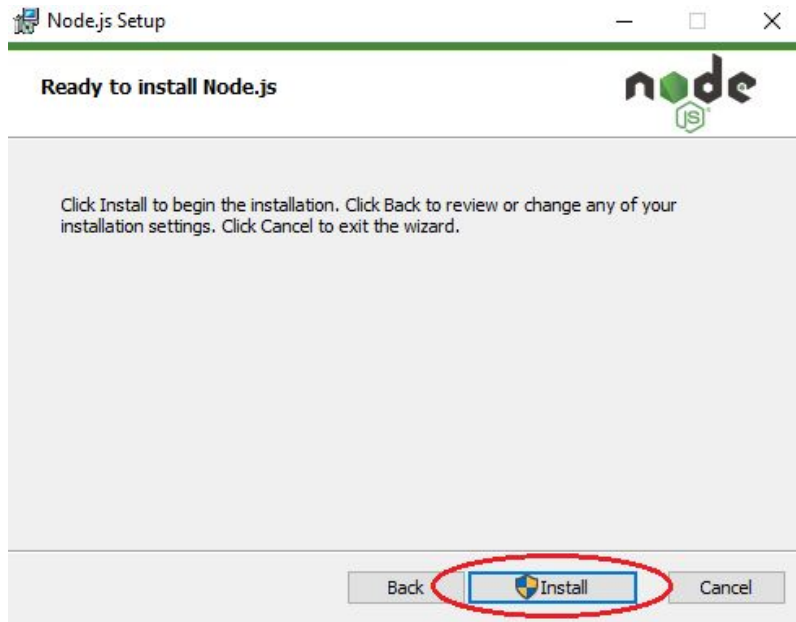
Instalação - TypeScript

- Instalação no Windows
 - **Importante:** escolher a opção **npm package manager** para instalar o gerenciador de pacotes do Node.js (esta opção não é marcada por padrão)



Instalação - TypeScript

- Instalação no Windows
 - Clicar em instalar



Instalação - TypeScript

- Instalação no Windows

- Para finalizar basta instalar o módulo do TypeScript para o Node.js
- Basta abrir a linha de comando do Windows
 - Clicar na pesquisa do windows (botão super)
 - Digitar **cmd**
 - Abra o **prompt de comando** da seguinte forma:
 - Necessita de permissão de administrador
 - Ao invés de clicar em **prompt de comando** normalmente, clica-se com o botão direito e depois em **executar como administrador**
- Digitar os seguintes comandos:
 - `npm install -g typescript`
 - `npm install -g readline-sync`
- Feche a linha de comando.
- Configure o projeto (mostrado a partir da slide [22](#)).

Instalação - TypeScript

- Instalação no Linux
 - Executar os seguintes comandos no terminal (precisa de permissão sudo):
 - `sudo apt-get update`
 - `sudo apt-get install nodejs`
 - `sudo apt-get install npm`
 - `sudo npm install -g typescript`
 - `sudo npm install -g readline-sync`
 - Pode acontecer o erro **EACCES, mkdir '/usr/local/lib/node_modules'**
 - Se acontecer, basta digitar os seguintes comandos (também precisa de permissão sudo):
 - `sudo apt-get install aptitude`
 - `sudo aptitude install nodejs-legacy`
 - Configure o projeto (mostrado a partir da slide [22](#)).

Instalação - TypeScript

- Criando um projeto (Windows e Linux)
 - Por causa da atualização do ECMAScript 6 (ES6), alguns problemas podem ocorrer se o projeto não for configurado corretamente.
 - Para que alguns recursos funcionem, temos que compilar dizendo que usaremos ES6.
 - No entanto, no momento de executar, só o que instalamos não funciona perfeitamente com o código compilado em ES6.
 - Assim, é necessário instalar mais alguns módulos do node para cada projeto que for ser criado.
 - Para configurar seu projeto corretamente, siga os passos a partir do Slide seguinte.

Criando um projeto (Windows e Linux)

1. Crie uma pasta onde será guardado o seu projeto (todos os arquivos do seu projeto estarão dentro dessa pasta). Ela será a pasta raiz do seu projeto.
2. Dentro dessa pasta, crie um arquivo chamado ***index.js*** (falaremos dele adiante).
3. Com o terminal (no Linux, ou prompt de comando do Windows), navegue até a pasta raiz do seu projeto. Usaremos o terminal para essa instalação.
 - Opcionalmente, você pode abrir a pasta com o Visual Studio Code e usar o terminal dele.
4. Digite o seguinte comando:
 - `npm init -y`
 - Esse comando, inicializará seu projeto como um pacote npm.
 - Esse comando também criará um arquivo chamado ***package.json*** semelhante ao arquivo mostrado a seguir.

Criando um projeto (Windows e Linux)

- Arquivo ***package.json***:

```
package.json x
{
  "name": "meu_projeto",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```


Criando um projeto (Windows e Linux)

5. Instale módulos necessários executando o seguinte comando:

- `npm install --save express morgan`
- Isso criará um arquivo **package-lock.json**, não o apague nem o altere.
- Esse comando irá alterar seu arquivo **package.json**, que ficará semelhante ao seguinte:

```
{
  "name": "meu_projeto",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.16.3",
    "morgan": "^1.9.0"
  }
}
```

Criando um projeto (Windows e Linux)

6. Instale os outros módulos necessários executando o seguinte comando:

- `npm install --save-dev babel-cli babel-preset-es2015 rimraf`
- Esse comando irá alterar seu arquivo **package.json**, que ficará semelhante ao seguinte:

```
{
  "name": "meu_projeto",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.16.3",
    "morgan": "^1.9.0"
  },
```

// continuação do arquivo

```
    "devDependencies": {
      "babel-cli": "^6.26.0",
      "babel-preset-es2015": "^6.24.1",
      "rimraf": "^2.6.2"
    }
  }
}
```

Criando um projeto (Windows e Linux)

7. Crie um arquivo chamado **.babelrc** com o seguinte conteúdo e o salve na pasta raiz do projeto (esse será um arquivo oculto, por isso inicia com ponto):

A screenshot of a code editor window. The title bar at the top shows ".babelrc" and a close button "x". The editor area contains the following JSON code:

```
{  
  "presets": ["es2015"]  
}
```

Criando um projeto (Windows e Linux)

8. Abra e edite o arquivo **package.json** da seguinte forma:

- Dentro das chaves do atributo “scripts” há uma linha semelhante a seguinte:
 - `"test": "echo \"Error: no test specified\" && exit 1"`
- Ao fim dessa linha, coloque uma vírgula (,) e aperte Enter.
- Abaixo dessa linha, insira essas duas linhas:
 - `"build": "rimraf dist/ && babel ./ --out-dir dist/ --ignore ./node_modules,./.babelrc,./package.json,./npm-debug.log --copy-files",`
 - `"start": "npm run build && node dist/index.js"`
- Salve o arquivo.
- Seu arquivo **package.json** deve ficar semelhante ao do Slide seguinte:

Criando um projeto (Windows e Linux)

```
{
  "name": "meu_projeto",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "build": "rimraf dist/ && babel ./ --out-dir dist/ --ignore ./node_modules,./babelrc,./package.json,./npm-debug.log --copy-files",
    "start": "npm run build && node dist/index.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.16.3",
    "morgan": "^1.9.0"
  },
  "devDependencies": {
    "babel-cli": "^6.26.0",
    "babel-preset-es2015": "^6.24.1",
    "rimraf": "^2.6.2"
  }
}
```

Criando um projeto (Windows e Linux)

9. Abra e edite o arquivo index.js que você criou no passo 2.

- Coloque o seguinte conteúdo no seu arquivo index.js e o salve:

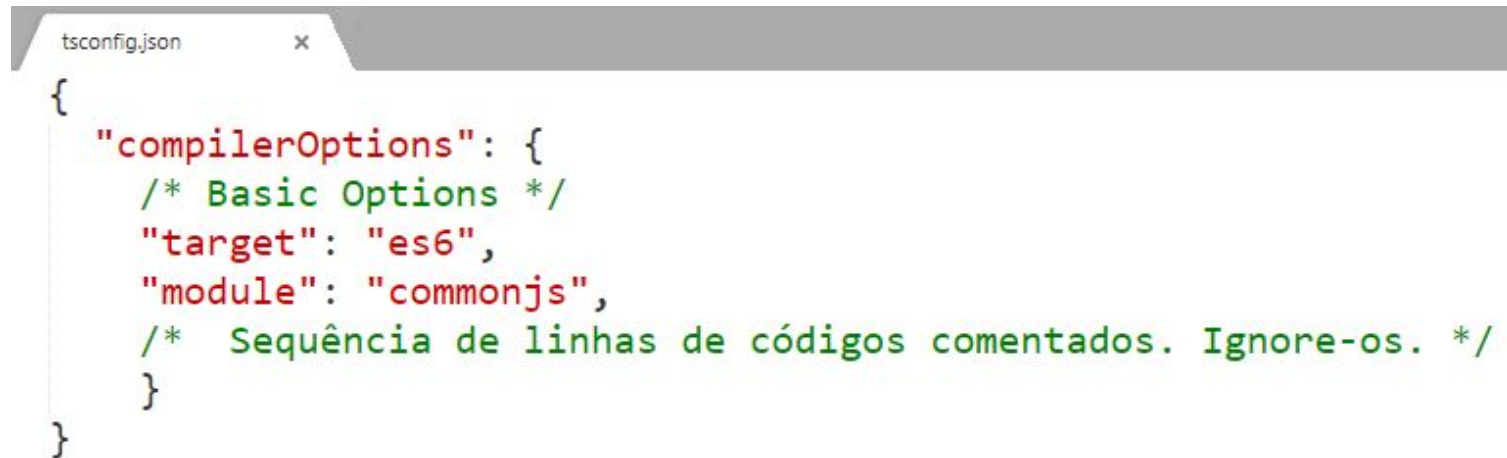
```
index.js x
require('babel-register')({
  presets: [ 'es2015' ]
});

require('./Principal');
/*
  Aqui é o nome do arquivo TypeScript que será executado, o "gatilho" da aplicação.
  Pode ser o nome que você desejar, você escolhe.
  Mas seu arquivo inicial do projeto terá que ter exatamente esse nome e a extensão TS.
  Por exemplo, nesse caso, seria o arquivo Principal.ts.
  Esse bloco de comentários é apenas explicativo, não precisa estar no seu arquivo index.js.
*/
```

Criando um projeto (Windows e Linux)

10. Execute o seguinte comando: `tsc --init`

- Isso criará um arquivo **tsconfig.json**. Abra esse arquivo.
- À frente do campo **“target”**:, temos o valor **“es5”**. Troque por **“es6”**.
- Salve o arquivo e feche-o.
- ***Esse é o último passo de configuração do seu projeto.***



```
tsconfig.json x
{
  "compilerOptions": {
    /* Basic Options */
    "target": "es6",
    "module": "commonjs",
    /* Sequência de linhas de códigos comentados. Ignore-os. */
  }
}
```

Execução do TypeScript

- Para compilar e executar os arquivos TS (TypeScript) é usada a linha de comando.
 - Prompt de comando no Windows.
 - Terminal no Linux.
 - Terminal (console) do VS Code.
 - Obviamente, a linha de comando deve estar aberta na pasta do projeto.
- Compilaremos e executaremos utilizando ECMAScript 6.
- Compilar o arquivo TS (transformando-o em um arquivo JavaScript).
 - `tsc -t ES6 Principal.ts`
 - Esse deve ser o arquivo principal da sua aplicação, o que está indicado no `index.js`.
 - São gerados automaticamente arquivos JS (JavaScript) com os mesmos nomes dos arquivos compilados.
- Executa-se o arquivo ***index.js*** utilizando Node.js.
 - `node index.js`

Perguntas?