

```

alexa@LAPTOP-ALE MINGW64 ~/Documents/2- Estudos/CC/4ª Fase/Desenvolvimento Web I
/Agosto/Configuração de ambiente/seminario_TypeScript (main)
$ npm init -y
Wrote to C:\Users\alexa\Documents\2- Estudos\CC\4ª Fase\Desenvolvimento Web I\Ag
osto\Configuração de ambiente\seminario_TypeScript\package.json:

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

alexa@LAPTOP-ALE MINGW64 ~/Documents/2- Estudos/CC/4ª Fase/Desenvolvimento Web I
/Agosto/Configuração de ambiente/seminario_TypeScript (main)
$ npm i typescript

added 1 package, and audited 2 packages in 11s

found 0 vulnerabilities

alexa@LAPTOP-ALE MINGW64 ~/Documents/2- Estudos/CC/4ª Fase/Desenvolvimento Web I
/Agosto/Configuração de ambiente/seminario_TypeScript (main)
$ npx tsc --init

Created a new tsconfig.json with:

  target: es2016
  module: commonjs
  strict: true
  esModuleInterop: true
  skipLibCheck: true
  forceConsistentCasingInFileNames: true

You can learn more at https://aka.ms/tsconfig

alexa@LAPTOP-ALE MINGW64 ~/Documents/2- Estudos/CC/4ª Fase/Desenvolvimento Web I
/Agosto/Configuração de ambiente/seminario_TypeScript (main)
$ npx tsc

```

**npm init -y:** Cria o arquivo package.json, nele adicionamos a seguinte linha ao nosso exemplo prático... `"type": "module",`

**npm i -D typescript:** Instala o TypeScript localmente em modo desenvolvedor (por isso o -D).

**npx tsc --init:** Um arquivo tsconfig.json será gerado. Para o nosso exemplo prático, o alteramos o conteúdo da linha "module" para `"module": "ESNext",`

**npx tsc:** pega todos os arquivos .ts e gera os arquivos correspondentes em .js