

# Passo a passo para criar um novo projeto typescript



---

## 1. INDEX

---

### 1. Introdução

1. [Objetivo.](#)
2. [Pre-requisitos.](#)
3. [benefícios.](#)

### 2. [Passo a passo para criar um novo projeto typescript.](#)

1. [Passo 01 - Criar a função rename\(\)](#)
2. [Passo 02 - Criar função ifExistProject\(\){}](#)
3. [Passo 03 - Criar as pasta do projeto;](#)
4. [Passo 04 - Criar workspace para o projeto do vscode na pasta raiz do projeto;](#)
5. [Passo 05 - Adicionar o código \*\*.html\*\* no arquivo `./src/html/index.html`;;](#)
6. [Passo 06 - Adicionar o código \*\*.css\*\* no arquivo `./src/css/defaulttheme.css`;;](#)
7. [Passo 07 - Adicionar o código \*\*.ts\*\* no arquivo `./src/ts/index.ts`;](#)
8. [Passo 08 - Executar o programa \*\*tsc-init\*\* para iniciar tudo que for preciso para o projeto;](#)
9. [Passo 09 - Instalar pacotes.](#)
10. [Passo 10 - Customizar o comportamento do webpack no arquivo \*\*webpack-config.js\*\*;](#)
11. [Passo11 - Customizar arquivo package.json;](#)
12. [Passo12 - Customizar arquivo tsconfig.json;](#)
13. [Versão completa \(my-tsc-init.sh\) dos 12 passos para iniciar um projeto typeScript.](#)

### 3. [Referências.](#)

### 4. [Histórico.](#)

---

## 2. CONTEÚDO

---

### 1. Introdução


#### 1. **Objetivo:**

1. Este documento descreve como configurar um novo projeto typescript com tudo que é preciso.
2. [\[← BACK\]](#)

#### 2. **Pre-requisitos:**

1. Os programs nodejs, typescript, tsc-init devem estar instalado globalmente. [Veja como...](#)
2. [\[← BACK\]](#)

### 3. Benefícios:

1. Ganhar temos ao iniciar um novo projeto pois os documentos oficiais não foram claros quando iniciei. [Veja por si mesmo...](#)
2. 

## 2. Passo a passo para criar um novo projeto typescript

### 1. Passo 01 - Criar a função rename()

#### 1. Código shellscript

```
# Função para renomeá arquivo e apagar o arquivo anterior
se existir
rename(){
#Parâmetros
# Chamada: rename param1 param2
fileName="$1";
fileNameOld="$2";

# begin
    if [ -f $fileNameOld ]; then
        # Se $fileNameOld já existe então apaga
        #      echo O arquivo $fileNameOld foi criado
        anteriormente!.
        #      echo Tecle ENTER para apaga-lo.
        #      read input
        rm $fileNameOld
    fi

    if [ -f $filename ]; then # Se $filename existe
        renomeia para $fileNameOld
        #      echo O arquivo $fileName foi criado
        anteriormente!.
        #      echo Tecle ENTER para renomear para
        $fileNameOld.
        #      read input
        mv $fileName $fileNameOld
    fi
# end
}
```

#### 2.

### 2. Passo 02 - Cria a função ifExistProject():

#### 1. Código shellscript

```

# Passo 02 - Retorna o nome do projeto ou aborta o script
ifExistProject(){

# Se o parâmetro 01 não existe então fazer nome do
project igual ./myproject
if [ -z $project ]; # -z indica que a string $pasta
existe e é vazia.
then
# echo -z $pasta retorno true
# read input
project="./myproject"
else
# echo -z $pasta retorno false
# read input
project="./$project"
fi

if [ -d $project ]; then # Se $pasta já existe então
aborta execução.
#echo O project $project foi criado anteriormente!.
Tecle ENTER para sair.
#read input
exit EEXIST # aborta o script

fi

}

```

## 2. BACK

### 3. Passo 03 - Criar as pasta do projeto:

#### 1. Código shellscript

```

# Passo 03 - Criar as pasta do projeto;
createProject(){

ifExistProject # Se o projeto já existe essa função
aborta

# Cria a pasta do project passada no parâmetro 01
mkdir $project

# Cria as pasta para os códigos fontes do project
mkdir ./"$project"/src      # Pasta raiz do código fonte
mkdir ./"$project"/src/html # Pasta html de entrada de
webpack
mkdir ./"$project"/src/css  # Pasta css de entrada de
webpack

```

```

mkdir ./"$project"/src/js      # Pasta de saída js de tsc
mkdir ./"$project"/src/ts      # Pasta de entrada ts de tsc
mkdir ./"$project"/dist        # Pasta destino do webpack
para ser publicada na web.

}

# PROGRAMA PRINCIPAL
# $1 é passado na execução de my-tsc-init.sh

project="$1"
createProject

```

## 2.

### 4. Passo 04 - Criar workspace para o projeto do vscode na pasta raiz do projeto.

#### 1. Código ShellScript

```

# Passo 04 - Criar workspace para o projeto do vscode na
pasta raiz do projeto;
createWorkspace(){

cat >."/"$project"/workspace.code-workspace"<<EOT
{
    "folders": [
        {
            "path": "."
        }
    ]
}
EOT

}

# PROGRAMA PRINCIPAL
# $1 é passado na execução de my-tsc-init.sh

project="$1"
createWorkspace

```

## 2.

### 5. Passo 05 - Adicionar o código **.html** no arquivo ./src/html/index.html.

#### 1. Código ShellScript

```
# Passo 05 - Adicionar o código .html no arquivo
./src/html/index.html;
createFileIndexHtml(){

# Criar arquivo inicial index.html
cat >"./"$project"/src/html/index.html"<<EOT

    <!DOCTYPE html>
        <html dir="ltr" lang="pt-br">

            <head>
                <meta http-equiv="content-type"
content="text/html; charset=utf-8" />

                <meta name="viewport"
content="width=device-width, initial-scale=1.0" />

                <title>Modelo de project
typescript</title>

                <meta name="createDate"
content="28/05/2021" />
                <meta name="createDateUpdate"
content="25/05/2021" />
                <meta name="description" content="Todos
project typescript deve seguir essa sequência ao iniciar..."
/>
                <meta name="keywords"
content="typescript,webpack" />

                <link type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css"
rel="stylesheet" />

            </head>

            <body>
                <p>Alo mundo</p>
            </body>

        </html>

    EOT

}

# PROGRAMA PRINCIPAL
# $1 é passado na execução de my-tsc-init.sh

project="$1"
```

```
createFileIndexHtml
```

2. 

6. Passo 06 - Adicionar o código **.css** no arquivo `./src/css/defaulttheme.css`;

1. Código ShellScript

```
# Passo 06 - Adicionar o código .css no arquivo
./src/css/defaulttheme.css;;
createFiledefaultThemeCss() {

cat >"./"$project"/src/css/defaulttheme.css"<<EOT
    html {
        scroll-behavior: smooth;
        /* scroll lento*/
    }

    body {
        margin: 0;
        font-family: Arial
    }

EOT
}

# PROGRAMA PRINCIPAL
# $1 é passado na execução de my-tsc-init.sh

project="$1"
createFiledefaultThemeCss
```

2. 

7. Passo 07 - Adicionar o código **.ts** no arquivo `./src/ts/index.ts`

1. Código ShellScript

```
# Passo 05 - Adicionar o código .ts no arquivo
./src/ts/index.ts;
createIndexTs(){

cat >"./"$project"/src/ts/index.ts"<<EOT
/**
 * Programa: Classe Print
 * Objetivo: Usada para imprimir um texto o console.
```

```

        * Criado em: 16/06/2021
        * Atualizado em: 16/06/2021
        */
class Print{
    /** Imprime um texto no console
    * @param s :String
    */
    public print(s:string){
        console.log(s);
    }
}

/**
 * Cria objeto p;
 */
let p = new Print();

/**
 * Imprime no console o texto : Alo Mundo
 */
p.print('Alo mundo');

EOT

}

# PROGRAMA PRINCIPAL
# $1 é passado na execução de my-tsc-init.sh

project="$1"
createIndexTs

```

## 2.

8. Passo 08 - Executar o programa **tsc-init** para iniciar tudo que for preciso para o projeto;

### 1. Código shellscript

```

# Passo 06 - Executar o programa tsc-init para iniciar
tudo que for preciso para o projeto;
execTscInit(){
    cd $project
    tsc-init
}

# PROGRAMA PRINCIPAL
# $1 é passado na execução de my-tsc-init.sh

project="$1"

```

```
execTscInit
```

2. 

## 9. Passo09 - Instalar pacotes.

### 1. Código shellscript

```
# Passo 09 - Instalar pacotes.
installPackages(){

    # Instala plugin a ser usado por alterWebpackConfig()
    npm install npm install --save-dev html-webpack-plugin
    # instalar plugin clean-webpack-plugin para deletar a
    pasta dist toda vez que que o comando npm run build for
    executado
    npm --save-dev install clean-webpack-plugin

}

# PROGRAMA PRINCIPAL
# $1 é passado na execução de my-tsc-init.sh

project="$1"
execTscInit
```

2. 

## 10. Passo 10 - Customizar o comportamento do webpack no arquivo **webpack-config.js**.

### 1. Código shellscript

```
# Passo10 - Customizar o comportamento do webpack no
arquivo webpack-config.js;
alterWebpackConfig(){

    echo "... ";
    echo "Adiciona const path = require('path') em
webpack.config.js;"
    echo "Adiciona a declaração \"const HtmlWebpackPlugin =
require('html-webpack-plugin')\" em webpack.config.js"

    subStrOrigem='module.exports = {'
    subStrDestino=" const path = require('path');\n const
HtmlWebpackPlugin = require('html-webpack-plugin'); \n\n
module.exports = {"
```



```

    fileNameOld="./webpack.config.js.ant"
    fileName="./webpack.config.js"

    rename "$fileName" "$fileNameOld"

    sed "s/$subStrOrigem/$subStrDestino/g" $fileNameOld >
$fileName

    echo ...

    echo Adiciona a propriedade "HtmlWebpackPlugin" em
module.exports = { webpack.config.js

    subStrOrigem='module.exports = {'
    subStrDestino="module.exports = \{plugins:\[new
HtmlWebpackPlugin\(\
{filename:'index.html',template:'.\src\html\index.html',i
nject:'body'\}\}\)\],"

    fileNameOld="./webpack.config.js.ant"
    fileName="./webpack.config.js"

    rename "$fileName" "$fileNameOld"

    sed "s/$subStrOrigem/$subStrDestino/g" $fileNameOld >
$fileName

    echo ...

    echo ...
    echo Alterar o arquivo webpack.config.js

    # Atualiza o arquivo de entrada de webpack.config.js
    subStrOrigem="entry: './index.ts'"
    subStrDestino="entry: './src/ts/index.ts'"
    fileNameOld="./webpack.config.js.ant"
    fileName="./webpack.config.js"
    rename "$fileName" "$fileNameOld"
    sed "s/$subStrOrigem/$subStrDestino/g" $fileNameOld >
$fileName
}
# PROGRAMA PRINCIPAL
# $1 é passado na execução de my-tsc-init.sh

project="$1"
alterWebpackConfig

```

2. 

11. Passo 11 - Customizar arquivo package.json;

## 1. Código shellscript

```
# Passo11 - Customizar arquivo package.json;
alterPackage(){
    echo ...

    echo Atualiza script "dev" do package.json
    subStrOrigem="webpack-dev-server --inline --hot"
    subStrDestino="webpack serve --mode development --env
development --hot --port 3000"
    fileNameOld="./package.json.ant"
    fileName="./package.json"
    rename "$fileName" "$fileNameOld"
    sed "s/$subStrOrigem/$subStrDestino/g" $fileNameOld >
$fileName

    echo ...
    echo Atualiza script "build" do package.json
    subStrOrigem="webpack -p"
    subStrDestino="webpack --mode='production'"
    fileNameOld="./package.json.ant"
    fileName="./package.json"
    rename "$fileName" "$fileNameOld"
    sed "s/$subStrOrigem/$subStrDestino/g" $fileNameOld >
$fileName
}

# PROGRAMA PRINCIPAL
# $1 é passado na execução de my-tsc-init.sh

project="$1"
alterPackage
```

## 2.

### 12. Passo 12 - Customizar arquivo tsconfig.json;

#### 1. Código shellscript

```
# Passo 12 - Customizar arquivo tsconfig.json;
alterTsconfig(){

    echo ...

    echo Adiciona a propriedade "outDir" em
"compilerOptions": {} do tsconfig.json

    subStrOrigem='"compilerOptions": {'
    subStrDestino='"compilerOptions": {"outDir":
"./src\\js", '
```

```
fileNameOld="./tsconfig.json.ant"
fileName="./tsconfig.json"

rename "$fileName" "$fileNameOld"

sed "s/$subStrOrigem/$subStrDestino/g" $fileNameOld >
$fileName


echo ...
}

# PROGRAMA PRINCIPAL
# $1 é passado na execução de my-tsc-init.sh

project="$1"
alterTsconfig
```


2. .

### 3. REFERÊNCIAS


1. [Documentação oficial typescript](#)
2. O programa [tsc-init](#) cria projeto typescript com tudo que é preciso para para o desenvolvimento.
3. [O que é shellscript](#)
4. [webpack x typescript](#)
5. <#>
6. <#>
7. .

### 4. HISTÓRICO

1. 28/05/2021

- ☒ Criar este documento baseado no modelo02.md ;
- ☒ Escrever tópico Objetivos;
- ☒ Escrever tópico Pre-requisitos
- ☒ Escrever tópico Benefícios
- 

2. 08/06/2021

- ☒ Escrever tópico Passo a Passo
  - ☒ Passo 01 - Função para renomeá arquivo e apagar o arquivo anterior se existir
  - ☒ # Passo 02 - Retorna o nome do projeto ou aborta o script ifExistProject(){}
  - ☒ # Passo 03 - Criar as pasta do projeto;createProject(){}
  - ☒ # Passo 04 - Criar workspace para o projeto do vscode na pasta raiz do projeto;createWorkspace(){}
  - ☒ # Passo 05 - Adicionar o código .html no arquivo ./src/html/index.html;createFileIndexHtml(){}
  - ☒ # Passo 06 - Adicionar o código .css no arquivo ./src/css/defaulttheme.css;createFiledefaultThemeCss() {}
  - ☒ # Passo 07 - Adicionar o código .ts no arquivo ./src/ts/index.ts;createIndexTs() {}
  - ☒ Passo 08 - Executar o programa tsc-init para iniciar tudo que for preciso para o projeto;execTsclnit(){}
  - ☒ # Passo 09 - Instalar pacotes.installPackages(){}
  - ☒ # Passo10 - Customizar o comportamento do webpack no arquivo webpack-config.js;alterWebpackConfig(){}
  - ☒ # Passo11 - Customizar arquivo package.json;alterPackage(){}
  - ☒ # Passo 12 - Customizar arquivo tsconfig.json;alterTsconfig(){}
- ☒ Escrever tópico Referências
- ☒ Atualizar o histórico deste documento.
- 

3. 15/06/2021

- ☐ Testar este documento depois após uma semana de concluído.

