



Desenvolvimento BACK-END I

Professor: Douglas Legramante

E-mail: douglas.legramante@ifro.edu.br



INSTITUTO FEDERAL
Rondônia
Campus Vilhena



JS

JavaScript

Aula 07 – Input

Input

O uso do input permite a entrada de dados dinâmicos na aplicação diretamente no console. Porém, esse não é um recurso nativo da linguagem.

Para isso, é necessário instalar o plugin **readline-sync** em nosso projeto.

Instalação do readline-sync

O processo de instalação é feito via linha de comando.

Mas antes, devemos criar a pasta do nosso projeto, pois realizaremos a instalação do plugin nessa pasta.

Uma dica de ouro:

Caso queira acessar diretamente a pasta, navegue até a pasta desejada via “Windows Explorer”, segure a tecla “SHIFT” e clique com o botão direito em algum lugar “vazio” da pasta. Será exibida a opção “Abrir a janela de comandos aqui”

Instalação do readline-sync

Abra o terminal (no VS code, CMD ou PowerShell) na pasta do projeto e digite o comando **npm init -y** e pressione enter.

npm init inicializa e o parâmetro **-y** é para ignorar o questionário feito pelo npm durante sua inicialização.

Se deu tudo certo, você verá uma tela como está ao lado.

E um novo arquivo, chamado **package.json** será criado automaticamente na pasta do projeto.

```
PS D:\Aulas\Back End I\Source> npm init -y
Wrote to D:\Aulas\Back End I\Source\package.json

{
  "name": "source",
  "version": "1.0.0",
  "description": "",
  "main": "01-FuncaoHelloWorld.js",
  "scripts": {
    "test": "echo \"Error: no test specified\"",
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

PS D:\Aulas\Back End I\Source> |
```

Instalação do readline- sync

Ainda no terminal, execute o comando **npm install readline-sync**

Esse comando instalará o **readline-Sync** no nosso projeto, além de criar outras pastas no diretório.

Pronto, a instalação está finalizada! Agora vamos configurá-lo.

```
PS D:\Aulas\Back End I\Source> npm install readline-sync
added 1 package, and audited 2 packages in 2s

found 0 vulnerabilities
PS D:\Aulas\Back End I\Source> |
```

Configurando do readline-sync

Vamos editar o arquivo
package.json:

1. Abra o arquivo no VS code, dando dois cliques sobre ele.
2. Adicione uma linha com o seguinte conteúdo:
"type": "module",
3. Salve as alterações.

```
{
  "name": "source",
  "version": "1.0.0",
  "description": "",
  "type": "module",
  "main": "01-FuncaoHelloWorld.js",
  ▶ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "readline-sync": "^1.4.10"
  }
}
```

Pronto, nosso plugin está configurado! Agora vamos testá-lo.

Testando o readline-sync

Crie um novo arquivo chamado `index.js`

Importe o `readline-Sync` adicionando o seguinte comando:

```
import entradaDados from 'readline-sync';
```

```
import entradaDados from 'readline-sync';

let nome = entradaDados.question('Digite seu nome: ');
console.log(`Olá, ${nome}.`);
```

Salve o código e execute o comando `node index.js` no terminal.

Atenção

Devido a uma limitação do Windows, caracteres acentuados podem não ser exibidos corretamente no prompt de comando (cmd). Esse não é um problema do **readline-sync** e sim do Windows. Para resolvê-lo, execute o comando **chcp 65001** no prompt de comando e a aplicação passará a exibir a acentuação corretamente.

Para saber mais

readlineSync – Java Script

<https://www.npmjs.com/package/readline-sync>

Cursos Gratuitos

Curso JavaScript Curso em Video

<https://www.cursoemvideo.com/curso/javascript/>

Curso JavaScript YouTube

<https://www.youtube.com/watch?v=McKNP3g6VBA>