

Entrada via console do NodeJS com uso de *callback*

Node.js permite execução do código JavaScript fora de um browser, permitindo a interação com dados em arquivos, criação de servidores Web e criação de aplicações de terminal de trabalho. Uma das maneiras com as quais o Node permite leitura de dados do usuário em ambientes offline como o VSCode é através de um módulo especial chamado `readline`, que permite criar uma interface de conexão com o *stream* de entrada.



[EXEMPLO] Programa que cumprimenta o usuário.

Uso de interface com stream de entrada.

```
1  const interface = require('readline').createInterface({
2    input: process.stdin,
3    output: process.stdout
4  });
5
6  interface.question('Qual o seu nome?\n', nome => {
7    console.log(`Olá, ${nome}!`);
8    interface.close();
9  });
```

A função `question(query, callback)` exibe a `query` para o usuário no stream de saída e assim que o usuário fornece a entrada desejada, a função de `callback` é aplicada tendo essa entrada como primeiro argumento.



[EXEMPLO] Programa para calcular a área de uma elipse com leitura de dados do usuário.

```
1  const interface = require('readline').createInterface({
2    input: process.stdin,
3    output: process.stdout
4  });
5
6  const farejaEli = (x, y, pi=3.14) => pi*x*y
7
8  interface.question('r1, r2? ', input => {
9    const raio = input.split(" ")
10   const r1 = parseFloat(raio[0])
11   const r2 = parseFloat(raio[1])
12   const resultado = farejaEli(r1,r2)
13   console.log(`Área da elipse é ${resultado}!`)
14   interface.close()
15 });
```



⚠ Para executar os códigos acima no VSCode, por exemplo, é preciso acessar o **TERMINAL** e executar o referido código através do comando a seguir (naturalmente, você deve estar no subdiretório onde seu arquivo `.js` está presente):

```
$ node nomedoarquivo.js
```