

Exercícios

Os exercícios abaixo devem ser solucionados com línguas Shell, como bash, nushell, zsh, fish, etc.

Na décima página, um exercício é proposto – *e erroneamente solucionado* – com o intuito de entender melhor os filtros em Shell scripting:

0. Suponha que exista um arquivo chamado `hotel.txt` com 100 linhas de dados. Imprima começando da linha 20 à linha 30 e armazene este resultado em um novo arquivo `hlist`.

```
$ tail +20 < hotel.txt | head -10 > hlist
```

BASH

```
$ open hotel.txt | lines | skip 20 | first 10 | save hlist
```

NUSHELL

Os exercícios a seguir são da página 44 em diante, e devem ser feitos escrevendo um script shell.

1. Adicione dois números, que são recebidos pela linha de comando como argumentos, e se esses dois números não são dados, mostre um erro e seu uso correto:

```
#!/usr/bin/env nu

# Add two numbers and return its sum.
def main [
  a: int # First number to add.
  b: int # Second number to add.
]: nothing -> int {
  $a + $b
}
```

NUSHELL

2. Retorne o maior número entre três argumentos dados pela linha de comando, e se três argumentos não forem dados, mostre um erro e seu uso correto.

```
#!/usr/bin/env nu

# Return the greatest number between the three given.
def main [
  a: int # First number to analyse.
  b: int # Second number to analyse.
  c: int # Third number to analyse.
]: nothing -> int {
  [ $a $b $c ] | sort | last
}
```

NUSHELL

3. Imprima a sequência 5, 4, 3, 2, 1 utilizando a repetição *while*:

```
#!/usr/bin/env nu

# Create the sequence 5, 4, 3, 2, 1 using the while loop.
```

NUSHELL

```
def main []: nothing -> list<int> {
  mut result = []

  mut i = 5
  while $i != 0 {
    $result = $result ++ [ $i ]
    $i = $i - 1
  }

  $result
}
```

4. Usando a palavra-chave `case` performe operações matemáticas básicas como adição (+), subtração (-), multiplicação (x), e divisão (/).

```
#!/usr/bin/env nu                                NUSHELL

# Using case (match) keyword to perform basic math operations such as addition
# (+), subtraction (-), multiplication (x), and division (/).
def main [
  a: int # First operand.
  operator: string # Operator.
  b: int # Second operand.
]: nothing -> int {
  match $operator {
    '+' => { $a + $b },
    '-' => { $a - $b },
    'x' => { $a * $b },
    '/' => { $a / $b },
    _ => { NaN }
  }
}
```