



Projeto Marvin

Etapa 03

Um infinito de números

Capítulo I

Instruções

- Somente esse material serve como referência. Não acredite em rumores.
- Fique atento, esse material pode mudar a qualquer hora antes da submissão.
- Os exercícios são cuidadosamente organizados em ordem de dificuldade, do mais fácil para o mais difícil. Esse mesmo raciocínio é aplicado para as correções, portanto não adianta completar um exercício mais difícil se os anteriores estão errados.
- Tenha certeza que possui as permissões necessárias para acessar arquivos e comandos.
- Você precisa seguir os procedimentos de envio para todas as atividades.
- Suas atividades serão corrigidas e avaliadas pelos seus colegas.
- Exercícios em Shell devem ser funcionais através do `/bin/bash`
- Você não pode deixar nenhum outro arquivo na pasta, a não ser o que foi expressamente pedido.
- Tem uma pergunta? Pergunte ao seu colega à direita. Caso contrário, tente o seu colega à esquerda.
- Seu guia de referência se chama Google / man / Internet
- Procure por conteúdo no Youtube para introduzir ao assunto.
- Examine os exemplos cuidadosamente. Eles estão sempre corretos e podem pedir por detalhes que não foram mencionados explicitamente na atividade.
- A função não deve ser chamada no arquivo, apenas definida.
- “No início, o universo foi criado. Isso irritou profundamente muitas pessoas e, no geral, foi encarado como uma péssima ideia”.
- Os arquivos devem ter compatibilidade

Capítulo II

Aleatório

_M	Exercício: 00
Aleatório	
Diretório de submissão: etapa03/ex00	
Arquivo a ser submetido: aleatorio.js	

- Escreva uma função chamada “aleatorio”, que retorna um número aleatório toda vez que executada.

```
function aleatorio() {  
  ...  
}  
console.log(aleatorio())
```

ao ser executado:

```
$>node aleatorio.js  
0.32344325432354  
  
$>node aleatorio.js  
0.80645170653298  
$>
```

Capítulo III

Exercício 01: Me fale um número

_M	Exercício: 01
Me fale um número	
Diretório de submissão: etapa03/ex01	
Arquivo a ser submetido: dezAVinte.js	

- Escreva uma função chamada “dezAVinte”, que retorna um número aleatório inteiro entre 10 e 20

```
function dezAVinte() {  
  ...  
}  
console.log(dezAVinte())
```

ao ser executado:

```
$>node dezAVinte.js  
11  
  
$>node dezAVinte.js  
20  
  
$>
```

Capítulo IV

Exercício 02: Inteiro

_M	Exercício: 02
Inteiro	
Diretório de submissão: etapa03/ex02/	
Arquivo a ser submetido: inteiro.js	

- Escreva uma função chamada “inteiro”, que recebe um número como argumento e retorna dizendo se esse número é inteiro ou não.

```
function inteiro(num) {  
  ...  
}  
console.log(inteiro("1111"))
```

ao ser executado:

```
$>node inteiro.js  
true
```

```
function inteiro(num) {  
  ...  
}  
console.log(inteiro(9234.22))
```

ao ser executado:

```
$>node inteiro.js  
false
```



verdadeiro ou falso

Capítulo V

Exercício 03: Soma

_M	Exercício: 03
Soma	
Diretório de submissão: etapa03/ex03/	
Arquivo a ser submetido: soma.js	

- Crie um função chamada "soma", que recebe três números como argumento e retorna a soma deles;

Capítulo VI

Exercício 04: Subtração

<code>_M</code>	Exercício: 04
Subtração	
Diretório de submissão: etapa03/ex04/	
Arquivo a ser submetido: subtracao.js	

- Escreva uma função chamada “subtracao”, que recebe dois números como argumento e retorna a subtração do primeiro menos o segundo número.

Capítulo VII

Exercício 05: Dividir

<code>_M</code>	Exercício: 05
Dividir	
Diretório de submissão: etapa03/ex05/	
Arquivo a ser submetido: dividir.js	

- Escreva uma função chamada “dividir”, que recebe dois números como argumento e retorna a divisão do primeiro pelo segundo número;

Capítulo VIII

Exercício 06: Vezes

<code>_M</code>	Exercício: 06
Vezes	
Diretório de submissão: etapa03/ex06/	
Arquivo a ser submetido: vezes.js	

- Escreva uma função chamada “vezes”, que recebe quatro números como argumento e retorna o resultado da multiplicação de todos esses números;

Capítulo IX

Exercício 07: Juntar

_M	Exercício: 07
Juntar	
Diretório de submissão: etapa03/ex07/	
Arquivo a ser submetido: juntar.js	

- Escreva uma função chamada “juntar”, que recebe dois números como argumento e retorna a concatenação deles;

```
function juntar(num1, num2) {  
  ...  
}  
console.log(juntar(12, 44))
```

ao ser executado:

```
$>node juntar.js  
1244
```

```
function juntar(num1, num2) {  
  ...  
}  
console.log(juntar(92, 0))
```

ao ser executado:

```
$>node juntar.js  
920
```

Capítulo X

Exercício 08: Raiz

<code>_M</code>	Exercício: 08
Raiz	
Diretório de submissão: etapa03/ex08/	
Arquivo a ser submetido: raizQuadrada.js	

- Escreva uma função chamada “raizQuadrada”, que recebe um número como argumento e retorna a raiz quadrada dele;