



Missão

Módulo 2 - Python

Preâmbulo: Neste Módulo 2, veremos como usar condições.

Versão: 2.0

Sumário

I	Uma palavra sobre esta Missão	2
II	Introdução	3
III	Instruções gerais	4
IV	Exercício 00: Sou Nulo?	5
V	Exercício 01: Sou Negativo?	6
VI	Exercício 02: Senhas	7
VII	Exercício 03: Multiplicação	8
VIII	Entrega e avaliação por pares	10

Capítulo I

Uma palavra sobre esta Missão

Bem-vindo(a)!

Você iniciará um Módulo desta Missão de programação de computadores. Nossso objetivo é apresentá-lo(a) ao código por trás do software que você usa diariamente e imergir na aprendizagem entre pares, o modelo educacional da 42.

Programação é sobre lógica, não matemática. Ela lhe dá blocos de construção básicos que você pode montar de inúmeras maneiras. Não existe uma única solução “correta” para um problema—sua solução será única, assim como a solução de cada um de seus colegas.

Rápido ou lento, elegante ou bagunçado, contanto que funcione, isso é o que importa! Esses blocos de construção formarão uma sequência de instruções (para cálculos, exibições, etc.) que o computador executará na ordem que você projetar.

Em vez de fornecer um curso onde cada problema tem apenas uma solução, nós o(a) colocamos em um ambiente de aprendizado entre pares. Você buscará elementos que possam ajudá-lo(a) a enfrentar seu desafio, refiná-los através de testes e experimentação e, finalmente, criar seu próprio programa. Discuta com os outros, compartilhe suas perspectivas, crie novas ideias juntos(as) e teste tudo você mesmo(a) para garantir que funcione.

A avaliação por pares é uma oportunidade fundamental para descobrir abordagens alternativas e identificar possíveis problemas em seu programa que você pode ter perdido (considere o quão frustrante pode ser uma falha de programa). Cada revisor abordará seu trabalho de forma diferente—como clientes com expectativas variadas— dando-lhe novas perspectivas. Você pode até formar conexões para futuras colaborações.

Ao final desta Missão, sua jornada será única. Você terá enfrentado diferentes desafios, validado diferentes projetos e escolhido caminhos diferentes dos outros—e isso é perfeitamente bom! Esta é uma experiência coletiva e individual, e todos(as) ganharão algo com ela.

Boa sorte a todos(as); esperamos que você aproveite esta jornada de descoberta.

Capítulo II

Introdução

O que este Módulo mostrará a você:

- Você aprenderá como fazer algumas condições.

Capítulo III

Instruções gerais

Salvo indicação em contrário, as seguintes regras se aplicam todos os dias desta Missão.

- Este documento é a única fonte confiável. Não confie em boatos.
- Este documento pode ser atualizado até uma hora antes do prazo de envio.
- As tarefas devem ser concluídas na ordem especificada. As tarefas posteriores não serão avaliadas, a menos que todas as anteriores sejam concluídas corretamente.
- Preste muita atenção aos direitos de acesso de seus arquivos e pastas.
- Suas tarefas serão avaliadas por seus(suas) colegas da Missão.
- Todas as tarefas de shell devem ser executadas usando `/bin/bash`.
- Você não deve deixar nenhum arquivo em seu espaço de trabalho de envio além daqueles explicitamente solicitados pelas tarefas.
- Tem alguma pergunta? Pergunte ao seu vizinho(a) à sua esquerda. Se não, tente seu vizinho(a) à sua direita.
- Toda resposta técnica que você precisa pode ser encontrada nas páginas `man` ou online.
- Lembre-se de usar o servidor do Discord!
- Leia os exemplos cuidadosamente, pois eles podem revelar requisitos que não são imediatamente óbvios na descrição da tarefa.
- Por Thor, por Odin! Use seu cérebro!!!

Capítulo IV

Exercício 00: Sou Nulo?

	Exercício : 00
	Sou Nulo?
	Pasta de entrega : <i>ex00</i> /
	Arquivos para entregar : <i>iszzero.py</i>
	Funções ou bibliotecas autorizadas : Todas

- Crie um programa chamado *iszzero.py*.
- Garanta que este programa seja executável (preste atenção às permissões do arquivo).
- Quando executado, o programa deve solicitar que o usuário insira um número.
- Se o número for igual a zero, o programa deve exibir: "Este número é igual a zero."
- Se o número não for igual a zero, o programa deve exibir: "Este número é diferente de zero."

```
?> ./iszzero.py
42
This number is different from zero.
?> ./iszzero.py
0
This number is equal to zero.
?>
```



Existe uma pequena linha para adicionar no início do seu arquivo para informar ao shell como interpretar o script. Pesquise por "Tornando um Script Python Executável".

Capítulo V

Exercício 01: Sou Negativo?

	Exercício : 01
Sou Negativo?	
Pasta de entrega : <i>ex01/</i>	
Arquivos para entregar : isneg.py	
Funções ou bibliotecas autorizadas : Todas	

- Crie um programa chamado **isneg.py**.
- Garanta que este programa seja executável.
- Quando executado, o programa deve solicitar que o usuário insira um número.
- Se o número for negativo, o programa deve exibir "Este número é negativo."
- Se o número for positivo, o programa deve exibir "Este número é positivo."
- Se o número for igual a zero, o programa deve exibir "Este número é positivo e negativo."

```
?> ./isneg.py
42
This number is positive.
?> ./isneg.py
-42
This number is negative.
?> ./isneg.py
0
This number is positive and negative.
?>
```

Capítulo VI

Exercício 02: Senhas

	Exercício : 02
	Acesso negado
	Pasta de entrega : <i>ex02/</i>
	Arquivos para entregar : password.py
	Funções ou bibliotecas autorizadas : Todas

- Crie um programa chamado **password.py**.
- Garanta que este programa seja executável.
- Defina uma variável contendo uma senha.

```
password = "Python is awesome"
```

- Quando executado, o programa deve solicitar que o usuário insira uma senha.
- Se a senha estiver correta, o programa deve exibir: "ACESSO CONCEDIDO".
- Se a senha estiver incorreta, o programa deve exibir: "ACESSO NEGADO".

```
?> ./password.py  
1234  
ACCESS DENIED  
?> ./password.py  
Python is awesome  
ACCESS GRANTED  
?>
```

Capítulo VII

Exercício 03: Multiplicação

	Exercício : 03
Multiplicação	
Pasta de entrega : <i>ex03/</i>	
Arquivos para entregar : mult.py	
Funções ou bibliotecas autorizadas : Todas	

- Crie um programa chamado **mult.py**.
- Garanta que este programa seja executável.
- Quando executado, o programa deve solicitar que o usuário insira 2 números.
- O programa deve:
 - Exibir se o resultado da multiplicação dos dois números é positivo, negativo ou zero.
 - Exibir o resultado da multiplicação.

```
$ ./mult.py
Enter the first number:
42
Enter the second number:
42
42 x 42 = 1764
The result is positive.

$ ./mult.py
Enter the first number:
78
Enter the second number:
-1
78 x -1 = -78
The result is negative.

$ ./mult.py
Enter the first number:
72
Enter the second number:
0
```

```
72 x 0 = 0
The result is positive and negative.
```

Capítulo VIII

Entrega e avaliação por pares

- Você deve ter a pasta `missao` na raiz do seu diretório inicial.
- Dentro da pasta `missao`, você deve ter uma pasta chamada `modulo2`.
- Dentro da pasta `modulo2`, você deve ter uma pasta para cada exercício.
- O Exercício 00 deve estar na pasta `ex00`, o Exercício 01 na pasta `ex01`, etc.
- Cada pasta de exercício deve conter os arquivos solicitados na tarefa.



Por favor, note que durante sua defesa, qualquer coisa que não estiver presente na pasta do dia não será verificada.