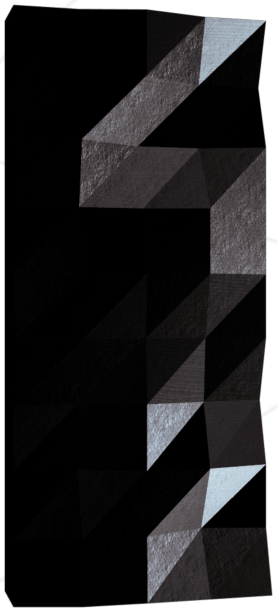

Missão

Módulo 3 - Python

Preâmbulo: Neste Módulo 3, veremos como usar loops.

Versão: 2.1

Sumário

Capítulo I

Uma palavra sobre esta Missão

Bem-vindo(a)!
Você começará um Módulo desta Missão de programação de computadores. Nosso objetivo é apresentar a você o código por trás do software que você usa diariamente e imersão em aprendizado entre pares, o modelo educacional da 42.

Programação é sobre lógica, não matemática. Ela fornece blocos de construção básicos que você pode montar de inúmeras maneiras. Não há uma única solução "correta" para um problema—sua solução será única, assim como as soluções de seus colegas.

Rápido ou lento, elegante ou bagunçado, contanto que funcione, 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 colocamos em um ambiente de aprendizado entre pares. Você procurará elementos que possam ajudá-lo(a) a enfrentar seu desafio, refiná-los por meio de testes e experimentação e, finalmente, criar seu próprio programa. Discuta com os outros, compartilhe suas perspectivas, apresente novas ideias juntos e teste tudo você mesmo 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 como uma falha de programa pode ser frustrante). Cada avaliador abordará seu trabalho de forma diferente—como clientes com expectativas variadas— dando a você 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 normal! Esta é uma experiência coletiva e individual, e todos ganharão algo com ela.

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

Capítulo II

Introdução

O que este Módulo mostrará a você:

- Você aprenderá como fazer alguns loops.

Capítulo III

Instruções gerais

A menos que especificado de outra forma, as seguintes regras se aplicam todos os dias desta Piscine.

- Este documento é a única fonte confiável. Não confie em boatos.
- Este documento pode ser atualizado até uma hora antes do prazo de entrega.
- As tarefas devem ser concluídas na ordem especificada. As tarefas posteriores não serão pontuadas, 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 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 uma pergunta? Pergunte ao seu vizinho à sua esquerda. Caso contrário, tente seu vizinho à sua direita.
- Toda resposta técnica de que você precisa pode ser encontrada nas páginas `man` ou online.
- Lembre-se de usar o servidor do Discord!
- Leia os exemplos atentamente, pois eles podem revelar requisitos que não são imediatamente óbvios na atribuição descrição.
- Por Thor, por Odin! Use seu cérebro!!!

Capítulo IV

Exercício 00: Até 25

	Exercício : 00
Vamos até 25!	
Pasta de entrega : $ex00/$	
Arquivos para entregar : `to25.py`	
Funções ou bibliotecas autorizadas : `Todos`	

- Crie um programa chamado `to25.py`.
- Garanta que o programa seja executável.
- O programa deve:
  - Aceitar a entrada do usuário, que será armazenada em uma variável numérica.
  - Use um loop para exibir todos os números de um número inserido até 25.
  - Se o número de entrada for maior que 25, exiba "Error"seguido por uma nova linha.

```
?> ./to25.py
Enter a number less than 25
45
Error
?> ./to25.py
Enter a number less than 25
20
Inside the loop, my variable is 20
Inside the loop, my variable is 21
Inside the loop, my variable is 22
Inside the loop, my variable is 23
Inside the loop, my variable is 24
Inside the loop, my variable is 25
?>
```

Use um loop while.

Capítulo V

Exercício 01: multiplication_table

	Exercício : 01
O retorno das tabuadas de multiplicação	
Pasta de entrega : $ex01/$	
Arquivos para entregar : `multiplication_table.py`	
Funções ou bibliotecas autorizadas : `Todos`	

- Crie um programa chamado `multiplication_table.py`.
- Garanta que o programa seja executável.
- O programa deve:
  - Aceita entrada do usuário, que será armazenada em uma variável numérica.
  - Exibe a tabuada para esse número (por exemplo, se a entrada for 2, exibe a tabuada de 2).

```
?> ./multiplication\_table.py
Enter a number
8
0 x 8 = 0
1 x 8 = 8
2 x 8 = 16
3 x 8 = 24
4 x 8 = 32
5 x 8 = 40
6 x 8 = 48
7 x 8 = 56
8 x 8 = 64
9 x 8 = 72
?>
```

Capítulo VI

Exercício 02: i_got_that

	Exercício : 02
Você entendeu?	
Pasta de entrega : $ex02/$	
Arquivos para entregar : `i_got_that.py`	
Funções ou bibliotecas autorizadas : `Todos`	

- Crie um programa chamado `i_got_that.py`.
- Garanta que o programa seja executável.
- O programa deve:
  - Use um loop `while` que aceite continuamente a entrada do usuário e responda com "Eu entendi! Mais alguma coisa?"após cada entrada.
  - O loop deve parar apenas quando o usuário inserir "STOP".

```
?> ./i_got_that.py
What you gotta say? : Hello
I got that! Anything else? : I like ponies
I got that! Anything else? : stop...
I got that! Anything else? : STOP
?>
```

Dê uma olhada em 'while' e 'break'.

Capítulo VII

Exercício 03: advanced_mult

	Exercício : 03
O Retorno do Retorno das Tabuadas de Multiplicação	
Pasta de entrega : $ex03/$	
Arquivos para entregar : `advanced_mult.py`	
Funções ou bibliotecas autorizadas : `Todos`	

- Crie um programa chamado `advanced_mult.py`.
- Garanta que o programa seja executável.
- O programa deve:
  - Exibir todas as tabuadas de multiplicação de 0 a 10 no seguinte formato:

```
?> ./advanced_mult.py
Table of 0: 0 0 0 0 0 0 0 0 0 0 0
Table of 1: 0 1 2 3 4 5 6 7 8 9 10
Table of 2: 0 2 4 6 8 10 12 14 16 18 20
Table of 3: 0 3 6 9 12 15 18 21 24 27 30
Table of 4: 0 4 8 12 16 20 24 28 32 36 40
Table of 5: 0 5 10 15 20 25 30 35 40 45 50
Table of 6: 0 6 12 18 24 30 36 42 48 54 60
Table of 7: 0 7 14 21 28 35 42 49 56 63 70
Table of 8: 0 8 16 24 32 40 48 56 64 72 80
Table of 9: 0 9 18 27 36 45 54 63 72 81 90
Table of 10: 0 10 20 30 40 50 60 70 80 90 100
?>
```

- Você só pode usar dois loops while.

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 `modulo3`.
- Dentro da pasta `modulo3`, 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.

`Observe que, durante sua defesa, qualquer coisa que não esteja presente na pasta do módulo não será verificada.`