



Discovery Piscine

Módulo 7 - Python

Preâmbulo: Neste Módulo, veremos como usar arrays e suas funções associadas.

Versão: 2.0

Sumário

I	Uma palavra sobre esta Missão	2
II	Introdução	3
III	Instruções gerais	4
IV	Exercício 00: parameter_matching	5
V	Exercício 01: count_it	6
VI	Exercício 02: string_are_arrays	7
VII	Exercício 03: append_it	8
VIII	Exercício 04: free_range	9
IX	Envio e avaliação por pares	10

Capítulo I

Uma palavra sobre esta Missão

Bem-vindo(a)!

Você começará o primeiro Módulo desta Missão de programação de computadores. Nossa objetivo é apresentar a você o código por trás do software que você usa diariamente e mergulhar no 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 existe 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, 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 procurará 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, apresente novas ideias juntos, 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 possa ter perdido (considere o quão frustrante uma falha de programa pode ser). Cada revisor abordará seu trabalho de forma diferente — como clientes com diferentes expectativas — dando a você novas perspectivas. Você pode até formar conexões para futuras colaborações.

Ao final desta Missão, sua jornada será única. Você terá enfrentado desafios diferentes, validado projetos diferentes 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 manipular arrays e suas funções associadas.

Capítulo III

Instruções gerais

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

- 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. 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.
- Todas as respostas técnicas de que você precisa podem ser encontradas nas páginas de `man` ou online.
- Lembre-se de usar o servidor do Discord!
- Leia os exemplos completamente, pois eles podem revelar requisitos que não são imediatamente óbvios na atribuição descrição.
- Por Thor, por Odin! Use seu cérebro!!!

Capítulo IV

Exercício 00: parameter_matching

	Exercício : 00
Identificando um Parâmetro	
Pasta de entrega : <i>ex00/</i>	
Arquivos para entregar : <i>parameter_matching.py</i>	
Funções ou bibliotecas autorizadas : All	

- Crie um programa chamado `parameter_matching.py`.
- Garanta que este programa seja executável.
- O programa deve:
 - Receber uma string como parâmetro.
 - Solicitar ao usuário para inserir uma palavra que corresponda ao parâmetro passado como um argumento, como mostrado no exemplo abaixo.
 - Exibir "Good job!" seguido por uma nova linha se a palavra inserida pelo usuário corresponder ao parâmetro passado.
 - Exibir "Nope, sorry..." seguido por uma nova linha se a palavra inserida pelo usuário não corresponder ao parâmetro passado.
 - Exibir "none" seguido por uma nova linha se o número de parâmetros passados for diferente de 1.

```
?> ./parameter_matching.py
none
?> ./parameter_matching.py "Hello"
What was the parameter? Bonjour
Nope, sorry...
?> ./parameter_matching.py "Hello"
What was the parameter? Hello
Good job!
?>
```

Capítulo V

Exercício 01: count_it

	Exercício : 01
Contando e Medindo Parâmetros	
Pasta de entrega : <i>ex01/</i>	
Arquivos para entregar : <i>count_it.py</i>	
Funções ou bibliotecas autorizadas : All	

- Crie um programa chamado `count_it.py`.
- Garanta que este programa seja executável.
- O programa deve:
 - Exibir "parameters:" seguido pelo número total de parâmetros passados como argumentos.
 - Para cada parâmetro, exibir o próprio parâmetro e seu comprimento, terminando com uma nova linha.
 - Se nenhum parâmetro for fornecido, exibir "none" seguido por uma nova linha.

```
?> ./count_it.py | cat -e
none$ 
?> ./count_it.py "Game" "of" "Thrones" | cat -e
parameters: 3$
Game: 4$
of: 2$
Thrones: 7$
?>
```



Desta vez, você deve usar um loop "for" em vez de um loop "while".

Capítulo VI

Exercício 02: string_are_arrays

	Exercício : 02
Entendendo Strings como Arrays	
Pasta de entrega : <i>ex02/</i>	
Arquivos para entregar : <code>string_are_arrays.py</code>	
Funções ou bibliotecas autorizadas : All	

- Crie um programa chamado `string_are_arrays.py` que receba uma string como parâmetro.
- Este programa deve ser executável.
- Quando executado, o programa deve exibir "z"para cada ocorrência do caractere "z"na string passada como parâmetro, seguido por uma nova linha.
- Se o número de parâmetros for diferente de 1, ou se não houver caracteres "z"na string, ele deve exibir "none"seguido por uma nova linha.

```
?> ./string_are_arrays.py | cat -e
none$
?> ./string_are_arrays.py "The character Z is not found in this string" | cat -e
none$
?> ./string_are_arrays.py "The character z is found in this string" | cat -e
z$
?> ./string_are_arrays.py "Zaz visits the zoo with Zazie" | cat -e
zzz$
?>
```



Strings também são compostas de caracteres individuais, assim como arrays. Tente!

Capítulo VII

Exercício 03: append_it

	Exercício : 03
Modificando Strings	
Pasta de entrega : <i>ex03/</i>	
Arquivos para entregar : append_it.py	
Funções ou bibliotecas autorizadas : All	

- Crie um programa chamado `append_it.py`.
- Garanta que este programa seja executável.
- O programa deve exibir cada parâmetro passado como um argumento, um por um, anexando-o com "ism".
- Se um parâmetro já terminar com "ism", ele deve ser ignorado e não exibido.
- Se não houver parâmetros, o programa deve exibir "none" seguido por uma nova linha.

```
?> ./append_it.py | cat -e
none$
?> ./append_it.py "parallel" "egoism" "human" | cat -e
parallelism$
humanism$
?>
```



Use `find`.

Capítulo VIII

Exercício 04: free_range

	Exercício : 04
Apenas um passo de distância!	
Pasta de entrega : <i>ex04/</i>	
Arquivos para entregar : free_range.py	
Funções ou bibliotecas autorizadas : All	

- Crie um programa chamado **free_range.py**.
- Garanta que este programa seja executável.
- Este programa deve receber 2 números como parâmetros.
- O primeiro número deve ser estritamente menor que o segundo.
- Você deve construir um array contendo todos os valores entre esses dois números.
- Você deve exibir o array usando a função **print**.
- Se o número de parâmetros for diferente de 2, o programa exibe 'none' seguido por uma nova linha.

```
?> ./free_range.py | cat -e
none$                                          
?> ./free_range.py 10 14 | cat -e
[10, 11, 12, 13, 14]$                        
?>
```



Use a função **range**.

Capítulo IX

Envio e avaliação por pares

- Você deve ter a pasta `missao` na raiz do seu diretório home.
- Dentro da pasta `missao`, você deve ter uma pasta chamada `module7`.
- Dentro da pasta `module7`, 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, durante sua defesa qualquer coisa que não esteja presente na pasta para o módulo não será verificado.