|  |  |
| --- | --- |
| **Data Processing with Python**  **Prof. Vagner Macedo** | **Uma imagem contendo Texto  Descrição gerada automaticamente** |

**Exercícios - Estrutura Sequencial**

1. Faça um Programa que peça um número e então mostre a mensagem *O número informado foi [número]*.

*xxxxxxxxxxxxxx*

1. Faça um Programa que converta metros para centímetros.
2. Faça um Programa que peça a temperatura em graus Farenheit, transforme e mostre a temperatura em graus Celsius.
   * C = (5 \* (F-32) / 9).

**Exercícios - Listas**

1. Faça um Programa que leia um vetor de 5 números inteiros e mostre-os.
2. Faça um Programa que leia 20 números inteiros e armazene-os num vetor. Armazene os números pares no vetor PAR e os números ímpares no vetor IMPARES. Imprima os vetores.
3. Faça um Programa que leia um vetor de 10 caracteres, e diga quantas consoantes foram lidas. Imprima as consoantes.

**Exercícios - Tuplas**

1. Crie uma tupla chamada cores contendo três cores de sua escolha. Em seguida, imprima cada cor em uma linha separada.
2. Dada a tupla ponto = (3, 4), calcule a distância do ponto à origem (0, 0) usando a fórmula matemática: 

,onde: x e y são as coordenadas do ponto.

1. Dada a tupla nomes = ('Data', 'Processing', 'with', 'Python', 'data'), conte quantas vezes o nome 'Data' aparece na tupla.

**Exercícios - Dicionários**

1. Crie um dicionário chamado estoque com itens de um supermercado e suas quantidades. Solicite ao usuário que digite o nome de um item. Exiba a quantidade desse item, se existir no estoque.
2. Crie um dicionário chamado agenda com nomes de pessoas e seus números de telefone. Permita que o usuário adicione novos contatos à agenda.
3. Crie dois dicionários, dicionario1 e dicionario2, com chaves e valores diferentes. Crie um terceiro dicionário, dicionario\_misturado, que contenha todas as chaves e valores de ambos dicionario1 e dicionario2.

**Exercícios - Sets**

1. Crie um conjunto com números inteiros e permita que o usuário adicione novos números. Remova todas as duplicatas do conjunto e imprima o conjunto resultante.
2. Crie dois conjuntos, conjunto1 e conjunto2, com elementos diferentes. Realize operações de união, interseção e diferença entre esses conjuntos e imprima os resultados.
3. Converta uma lista em um conjunto e, em seguida, converta o conjunto de volta para uma lista. Imprima a lista resultante para verificar se a conversão foi bem-sucedida.

**Exercícios - Funções**

1. Crie uma função chamada calcular\_media que recebe uma lista de números como argumento e retorna a média desses números.
2. Crie uma função chamada verificar\_paridade que recebe um número como argumento e retorna se o número é par ou ímpar.
3. Crie uma função lambda que recebe dois números como argumentos e retorna o primeiro número elevado à potência do segundo número.

**Exercícios - Pacotes**

1. Crie um módulo Python com uma função que imprime uma mensagem. Importe e execute essa função em outro script Python.
2. Crie um pacote Python com dois módulos diferentes. Importe e utilize funções de ambos os módulos em outro script Python.
3. Utilize o pacote requests para fazer uma requisição HTTP em um script Python. Retorne o status\_code.

**Exercícios - Arquivos e I/O**

1. Leia o conteúdo de um arquivo de texto e imprima-o na tela.
2. Solicite ao usuário que digite algumas linhas de texto e grave essas linhas em um arquivo de texto.

**Exercícios - Tempo**

1. Obtenha a data atual e imprima-a na tela.
2. Obtenha a data atual e imprima-a na tela no formato dia/mês/ano.
3. Agende uma tarefa para ser executada daqui a 10 minutos e imprima uma mensagem quando ela for executada.

**Exercícios - Regex**

1. Valide se um endereço de e-mail está em um formato válido usando expressões regulares.

[regex101: build, test, and debug regex](https://regex101.com/)

1. Extraia todos os números de uma string usando expressões regulares.
2. Verifique se uma string está no formato de data dd/mm/aaaa usando expressões regulares.