

ORIENTAÇÕES

1 OBJETIVO

Criar algoritmos na linguagem de programação Python, utilizando os materiais disponíveis nas Rotas de Aprendizagem e exemplos das Aula Práticas.

2 MATERIAL UTILIZADO

A Atividade Prática de Linguagem de Programação será realizada com base nas 6 Rotas de Aprendizagem e na linguagem de programação PYTHON. Poderá o Google Colab ou Anaconda ou PyCharm ou Visual Code.

3 ORIENTAÇÕES GERAIS

- Para cada exercício, coloque no seu relatório o **ENUNCIADO** do mesmo e coloque como resposta o seu **código COMPLETO**. Desde a primeira até a última linha que você digitou;
- Coloque no seu código **COMENTÁRIOS** explicando **COM SUAS PALAVRAS** o que ele faz (veja o exemplo do documento modelo);
- Google Colab ou PyCharm ou Jupyter: para inserir os códigos nas respostas, faça um **CTRL+C/CTRL+V** do código criado por você na ferramenta, e cole-o no documento do Word. Pode ainda utilizar o botão direito do mouse para escolher a formatação adequada no word. Assim, o código já virá colorido, organizado e indentado, facilitando a correção da sua atividade;
- No AVA existe um modelo em WORD para você utilizar. Deverá submeter o seu **arquivo em PDF**, basta apertar em salvar como PDF no Word;
- Além do seu algoritmo, você deverá colocar uma captura de tela do seu código funcionando. Capture o terminal mostrando o seu código funcionando e imprimindo os dados solicitados na tela (veja o exemplo do modelo).

CUIDADO! ✓ Em programação, não existem dois códigos exatamente iguais. Cada programador organiza seu código de uma forma diferente, declara variáveis com nomes diferentes, faz comentários diferentes, gera mensagens aos usuários distintas, etc. Por este motivo, e como a atividade é INDIVIDUAL, não serão aceitos dois algoritmos idênticos entre alunos (ou iguais à Internet). Caso o corretor observe respostas iguais, elas serão consideradas como PLÁGIO e arquivo em FORMATO ERRADO será atribuída a NOTA ZERO na questão;

4. COMO SE DARÁ A CORREÇÃO DA ATIVIDADE?

A Atividade Prática é composta por 3 questões. Para que você ganhe nota máxima em cada exercício, você precisará cumprir os três requisitos básicos explicados nas **ORIENTAÇÕES GERAIS**:

- Apresentar seu algoritmo completo, indentado e organizado;
- **CTRL+C/CTRL+V** do código criado por você na ferramenta, e cole-o no documento do Word. (Print do código não serão aceitas)
- Explicar seu código através de comentários;

- Colocar uma IMAGEM com o terminal rodando e mostrando o que cada exercício pede. No modelo de relatório da disciplina você encontrará um exemplo de exercício para um melhor entendimento. Caso você desenvolva seu código corretamente e funcional, porém não faça os comentários nem coloque uma imagem dele funcionando no terminal, terá sua nota severamente prejudicada.

5. EXEMPLO DE RESOLUÇÃO

Exercício 01 - Realizar uma classe veículo que apresente o tipo do carro, modelo, fabricante e quantidade de passageiros. Além disso, utilizar os métodos GETTER e SETTER para incluir as classes Modelo retornando o modelo e o fabricante do carro e outra classe Quantidade de Passageiro no carro.

Resposta do exercício 01:

Não esqueça de explicar o código com comentários!! CTRL+C/CTRL+V da ferramenta!! Se desejar, defina as páginas do documento para PAISAGEM para melhor inserir os códigos.

CTRL+C/CTRL+V DO CÓDIGO:

```
class Veiculo:
    def __init__(self, tipo, modelo, fabricante, qtd_passageiro):
        self.tipo = tipo
        self.modelo = modelo
        self.fabricante = fabricante
        self.qtd_passageiro = qtd_passageiro
MyBus= Veiculo('onibus', "JS20", "Rota", 35)
MyCar = Veiculo('carro', modelo="Hillux", fabricante="Toyota", qtd_passageiro=5)
print( MyBus.tipo )
print( MyBus.fabricante )
print( MyBus.modelo )
print( MyBus.qtd_passageiro )
```

Foto do terminal sendo compilado sem erros:



```
[3] print( MyBus.tipo )
print( MyBus.fabricante )
print( MyBus.modelo )
print( MyBus.qtd_passageiro )

onibus
Rota
JS20
35

print( MyCar.tipo )
print( MyCar.fabricante )
print( MyCar.modelo )
print( MyCar.qtd_passageiro )

carro
Toyota
Hillux
5
```

6 EXERCÍCIOS A SEREM SOLUCIONADOS

Exercício 1 – Desenvolver uma **classe calculadora** que faça operações matemáticas utilizando dois números inteiros, sendo os **dois últimos números de seu RU**. Os dois números inteiros devem ser solicitados para o usuário digitar. Se o seu **RU** for **zero**, substituí-lo(s) pelo **número 5**. Sendo as possíveis operações matemáticas: **soma(+)**, **subtração(-)**, **multiplicação(*)**, **divisão(/)**, **expoente (^)**, **resto(%)** e **raiz quadrada da soma dos dois números** (**$\sqrt{\text{Num1} + \text{Num2}}$**). Além destas funcionalidades, o algoritmo deverá ter um **MENU** que possibilite ao usuário escolher qual o tipo de operação que se deseja realizar e que possibilite ao usuário a digitar os dois números. Apresentar todas as operações matemáticas da calculadora funcionando!

Exercício 2 – Dada uma determinada equação linear **$y = ax + bx - c$** , sendo que os valores para **a**, **b** e **c** serão os **três primeiros números de seu RU** (**a = NUM1**, **b = NUM2**, **c = NUM3**). Caso, **algum número do RU seja igual a zero**, substituí-lo(s) pelo **número 5**. Além disso, será preciso criar um vetor aleatório de tamanho 10, onde cada posição do vetor conterá os valores de **x para a equação linear**. Feito isto, fazer um gráfico para mostrar os pontos obtidos pela equação linear dentro do plano cartesiano. Por fim, nomear os eixos Y e X do gráfico, colocar cores diferentes para os pontos e colocar legenda. ***Dica:** você vai ter no total, 10 pontos no seu plano cartesiano, ou seja, 10 pontos serão ilustrados no gráfico.*

Exercício 3 – Realizar o upload do arquivo **STORES.csv**. Renomear todas as colunas do arquivo STORES.csv, onde os respectivos nomes sejam compactados (**Exemplo: Daily_Customer_Count foi renomeado para Visitantes**). Após isto, para se analisar o desempenho das lojas de supermercado/mercado do arquivo STORES.csv encontre os valores **mínimo**, **máximo**, **médio** e **desvio padrão** das seguinte colunas: **"Items_Available"**; **"Daily_Customer_Count"**; e **"Store_Sales"**. Posto isto, realizar três gráficos com as seguintes informações: Items_Available, Daily_Customer_Count e Store_Sales. Não se esqueça de colocar: nomes para os eixos Y e X do gráfico, colocar cores diferentes para os pontos e colocar legenda.

Algumas informações extras sobre a tabela do arquivo **STORES.csv**:

- *ID da loja: (Índice) ID da loja específica.*
- *Store da loja: Área Física da loja em pátio.*
- *Itens Avaliados: Número de itens diferentes disponíveis na loja correspondente.*
- *Contagem diária de clientes: Número de clientes que visitaram as lojas em média ao longo do mês.*
 - *Histórico de vendas: Vendas em (US\$) que as lojas realizaram.*