Exercicios_01_respostas

September 26, 2024

1 Python Básico

1.1 Exercícios Referentes ao Notebook LDS_Python_1 e LDS_Python_2

Para que estes exercícios sejam úteis ao aprendizado, antes de iniciar, desative no menu o uso de assitência por AI.

Iremos exercitar os a seguir os temas desenvolvidos na aula de introdução ao Python, cobrindo sintaxe, comentários, variáveis e tipos de dados.

1.1.1 1. Exercícios de Revisão

a) A indentação em Python é um elemento crucial da linguagem, pois define a estrutura do código. Em vez de usar chaves {} ou outros delimitadores, como em algumas outras linguagens de programação, Python usa a indentação (geralmente com 4 espaços) para agrupar blocos de código. Ela é obrigatória para definir instruções como condicionais, laços e funções. **Exemplo**:

```
if 5 > 2:
  print("Cinco é maior que dois!")
```

Se a indentação não for feita corretamente, o código resultará em erros de sintaxe ou funcionará de maneira incorreta. O código abaixo não está rodando, corrija o erro para que apresente o resultado esperado:

[1]:
```
# Solução
x = 3
y = 5

if x < y:
  print("x is smaller than y")
else:
  print("x is greater than or equal to y")
```

```
x is smaller than y
```

b) Erros de Sintaxe podem acontecer por vários motivos, incluindo digitação. Corrija os erros de digitação no código a seguir para que apresente o resultado abaixo:

```
A soma de 25 + 5 é igual a 30?
True
```

[2]:
```
X = 25
b = X + 5
print("A soma de 25 + 5 é igual a 30?")
b == 30
```

```
A soma de 25 + 5 é igual a 30?
```

[2]: `True`

c) Comentários ajudam a deixar o código mais fácil de entender pois algumas decisoes ou mesmo a lógica podem se tornar complexas ao desenvolver um programa maior. Um comentario inicia com o símbolo #, complete o codigo abaixo, conforme o modelo, inserindo comentarios:

```
# Este programa calcula a soma de dois numeros

a = 5  # atribui 5 a variavel a
b = 3  # atribui 3 a variavel b
sum = a + b  # Soma a e b, guardando o resultado na variavel sum

print("The sum is:", sum)  # imprime a variavel sum
```

[3]:
```
# Solução
a = 2  # A variável 'a' recebe o valor 2
b = a + 3  # A variável 'b' recebe o valor de 'a' (2) somado a 3, ou seja, b = 5
oper = a**b  # A variável 'oper' armazena o valor de 'a' elevado à potência de␣
 ↪'b', ou seja, 2^5 = 32

print("Resultado:", oper)  # Exibe na tela o texto "Resultado:" seguido do␣
 ↪valor de 'oper', que é 32
```

```
Resultado: 32
```

d) Muitas vezes iremos comentar um bloco de código, pois não desejamos que ele seja executado para um teste por exemplo. Podemos colocar # no início de todas as linhas, selecionando o código e pressionando *CONTROL* / Use o atalho *CONTROL* / para remover o cometário do código abaixo e execute o código.

[4]:
```
# Solução

a = 2
b = a + 7
oper = a**b

print("Resultado:", oper)
```

```
Resultado: 512
```

1.2 2. Problemas

a) Você precisa implementar um código em Python que use o valor principal (capital) inicial, a taxa de juros por período e o tempo em que o dinheiro é emprestado ou investido. Com base nesses valores, o programa deve calcular os juros simples e o montante total ao final do período.

[5]:
```
# Solução:
capital = 10_250 # R$ 10.250
taxa_juros = 7 # valor em percentual 10%
tempo = 3 # tempo investido, por 3 meses neste caso

# Converter para decimal
taxa_juros = taxa_juros / 100

# Calcula os juros simples
juros_simples = capital * taxa_juros * tempo

# Calcula o montante total (capital + juros)
montante_total = capital + juros_simples

# Exibe o resultado
print(f"Juros simples: {juros_simples:.2f}")
print(f"Montante total ao final do período: {montante_total:.2f}")
```

```
Juros simples: 2152.50
Montante total ao final do período: 12402.50
```

b) O funcionário precisa dividir um número de maçãs igualmente entre os alunos que vieram jantar no RU e verificar quantas maçãs sobram ao final. Ele conta que existem 67 maçã na bandeja e o total de 13 alunos na fila. Complete o bloco de código em branco, usando as operações // e %, para resolver o problema.

[6]:
```
#Solução
# Quantidade de maçãs e pessoas
macas = 67
pessoas = 13

# Divisão inteira (quantas maçãs cada pessoa recebe)
macas_por_pessoa = macas // pessoas

# Resto da divisão (quantas maçãs sobram)
sobras = macas % pessoas

# Exibindo o resultado
print("Cada pessoa recebe", macas_por_pessoa, "maçãs.")
print("Sobraram", sobras, "maçãs.")
```

```
Cada pessoa recebe 5 maçãs.
Sobraram 2 maçãs.
```

c) Para entrar dados em seu programa no tempo de execução é possível usar a função *input*:

```
variavel = input()
```

Imagine que você está implementando um sistema para verificar se os alunos de uma turma estudantil passaram na disciplina ou não. Para isso solicite que o usuário insira as notas das 4 provas realizadas por um estudante e calcule a média. Após isso, emita uma resposta booleana (True ou False) se o estudante passou na disciplina pensando que a média mínima para aprovação é que seja pelo menos 5.

[7]:
```
# Solução
print("Entre com as notas parciais:")
p1=float (input('P1: ')) #float garante que o dado inserido será do tipo float.
 ↪O default é string.
p2=float (input('P2: '))
p3=float (input('P3: '))
p4=float (input('P4: '))
n=p1+p2+p3+p4
n/=4
print("Media=",n)
print("Aprovado:",n>=5)
```

```
Entre com as notas parciais:

P1:  5
P2:  5
P3:  5
P4:  6

Media= 5.25
Aprovado: True
```