

Calculadora em Python

Autor: Ana Siqueira

Trabalho Prático: DGT2817 LÓGICA, ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

Microatividade 1:

Objetivo: Descrever a utilização das estruturas de condição if e else em Python

Foi criado um código para verificar a temperatura. Caso a temperatura fosse menor que 30, seria considerada amena. Maior que isso, seria considerada quente.

```
estruturas_condicao1.py > ...
1  temperatura = 29
2
3  if temperatura < 30:
4      print('A temperatura hoje está amena')
5  else:
6      print('Hoje está fazendo calor')
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_condicao1.py
A temperatura hoje está amena
PS D:\Cursos\Faculdade\Microatividades> python d:/Cursos/Faculdade/Microatividades/estruturas_condicao1.py
A temperatura hoje está amena
PS D:\Cursos\Faculdade\Microatividades> 
```

```
estruturas_condicao1.py > ...
1  temperatura = 31
2
3  if temperatura < 30:
4      print('A temperatura hoje está amena')
5  else:
6      print('Hoje está fazendo calor')
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_condicao1.py
A temperatura hoje está amena
PS D:\Cursos\Faculdade\Microatividades> python d:/Cursos/Faculdade/Microatividades/estruturas_condicao1.py
A temperatura hoje está amena
PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_condicao1.py
Hoje está fazendo calor
PS D:\Cursos\Faculdade\Microatividades> 
```

Microatividade 2

Objetivo: Descrever a utilização da estrutura de condição else if (elif) em Python

Foi criado um código para analisar o nível de experiência de um profissional. Abaixo de 2 anos é considerado júnior, entre 2 e 5 é considerado pleno e acima disso é considerado sênior.

```
estruturas_condicao2.py > ...
1 tempoExperiencia = 5
2
3 if tempoExperiencia < 2:
4     print('Nível de conhecimento júnior.')
5 elif tempoExperiencia > 2 and tempoExperiencia < 5:
6     print('Nível de conhecimento pleno.')
7 else:
8     print('Nível de conhecimento sênior.')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_condicao2.py
Nível de conhecimento sênior.
PS D:\Cursos\Faculdade\Microatividades> █
```

```
estruturas_condicao2.py > ...
1 tempoExperiencia = 1
2
3 if tempoExperiencia < 2:
4     print('Nível de conhecimento júnior.')
5 elif tempoExperiencia > 2 and tempoExperiencia < 5:
6     print('Nível de conhecimento pleno.')
7 else:
8     print('Nível de conhecimento sênior.')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_condicao2.py
Nível de conhecimento sênior.
PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_condicao2.py
Nível de conhecimento júnior.
PS D:\Cursos\Faculdade\Microatividades> █
```

```
estruturas_condicao2.py > ...
1  tempoExperiencia = 3
2
3  if tempoExperiencia < 2:
4      print('Nível de conhecimento júnior.')
5  elif tempoExperiencia > 2 and tempoExperiencia < 5:
6      print('Nível de conhecimento pleno.')
7  else:
8      print('Nível de conhecimento sênior.')

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_condicao2.py
Nível de conhecimento sênior.
PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_condicao2.py
Nível de conhecimento júnior.
PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_condicao2.py
Nível de conhecimento pleno.
PS D:\Cursos\Faculdade\Microatividades> █
```

Microatividade 3

Objetivo: Descrever a utilização da estrutura de repetição while em Python

Demonstração do loop while, onde continua pedindo um número enquanto o usuário não digita 0.

```
estruturas_repeticao1.py > ...
1  entrada_idade = ""
2
3  while entrada_idade != "0":
4      entrada_idade = input("Digite um número qualquer ou 0 para sair: ")
5      if entrada_idade != "0":
6          print("Número digitado: " + entrada_idade)
7
8  print("Programa encerrado.")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_repeticao1.py
Digite um número qualquer ou 0 para sair: 10
Número digitado: 10
Digite um número qualquer ou 0 para sair: 20
Número digitado: 20
Digite um número qualquer ou 0 para sair: 5
Número digitado: 5
Digite um número qualquer ou 0 para sair: 4
Número digitado: 4
Digite um número qualquer ou 0 para sair: 3
Número digitado: 3
Digite um número qualquer ou 0 para sair: 0
Programa encerrado.
PS D:\Cursos\Faculdade\Microatividades> █
```

Microatividade 4

Objetivo: Descrever a utilização da estrutura de repetição for em Python

Demonstração do laço for, onde primeiro lista todos os caracteres contidos na variável de texto, e depois números de 1 a 10. Foi utilizado o "range(1, 11)" pois o último número não é contabilizado.

```
estruturas_repeticao2.py > ...
1  texto = 'Olá, laço for'
2
3  for item in texto:
4      print('Caractere: ' + item)
5
6  for numero in range(1, 11):
7      print('Número do intervalo: ' + str(numero))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/estruturas_repeticao2.py
Caractere: O
Caractere: l
Caractere: á
Caractere: ,
Caractere: 
Caractere: l
Caractere: a
Caractere: ç
Caractere: o
Caractere: 
Caractere: f
Caractere: o
Caractere: r
Número do intervalo: 1
Número do intervalo: 2
Número do intervalo: 3
Número do intervalo: 4
Número do intervalo: 5
Número do intervalo: 6
Número do intervalo: 7
Número do intervalo: 8
Número do intervalo: 9
Número do intervalo: 10
PS D:\Cursos\Faculdade\Microatividades> █
```

Microatividade 5

Objetivo: Descrever a utilização de funções em Python

Demonstração de uma função simples que imprime "Olá, funções em Python" no terminal.

```
funcoes1.py > ...
1 def imprimir_variavel():
2     texto = 'Olá, funções em Python'
3     print(texto)
4
5 imprimir_variavel()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/funcoes1.py
Olá, funções em Python
PS D:\Cursos\Faculdade\Microatividades> 
```

Microatividade 6

Objetivo: Descrever a utilização argumentos de funções no Python

Função criada para analisar se o cliente é um usuário comum ou um administrador, e mostrar mensagens diferentes para cada um.

```
funcoes2.py > ...
1 def loginUsuario(perfil):
2     if perfil.lower() == 'admin':
3         print('Bem-vindo, Administrador')
4     else:
5         print('Bem-vindo, Usuário')
6
7 loginUsuario('Admin')
8 loginUsuario('admin')
9 loginUsuario('User')
10 loginUsuario('usuário')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/funcoes2.py
Bem-vindo, Administrador
Bem-vindo, Administrador
Bem-vindo, Usuário
Bem-vindo, Usuário
PS D:\Cursos\Faculdade\Microatividades> 
```

Atividade Prática

Objetivo: Refatorar o código de uma calculadora, acrescentando ao mesmo alguns dos recursos da linguagem Python vistos ao longo das microatividades.

Foi criado um código de uma calculadora em termina, aonde aceita como entrada dois números e uma operação. Essa calculadora executa operações básicas como adição, subtração, multiplicação e divisão. Ela identifica a operação digitada por símbolo ou nomes, e executa a função correspondente. Também foi tratados erros comuns, como divisão de algum número por 0 e digitação de algum caractere inválido. Além disso, devido ao loop while, o usuário consegue continuar fazendo cálculos até que decida sair da aplicação.

```

calculadora_v2.py X
calculadora_v2.py > ...
1  #Controle
2  saida = ""
3
4  #Operações Básicas
5  def adicao(a, b):
6      return a + b
7
8  def subtracao(a, b):
9      return a - b
10
11 def multiplicacao(a, b):
12     return a * b
13
14 def divisao(a, b):
15     if b == 0:
16         print('Não foi possível realizar a divisão por 0')
17     else:
18         return a / b
19
20 #Função Calculadora
21 def calculadora(num1, num2, operacao):
22     operacao = operacao.lower()
23     resultado = None
24
25     if operacao in ['+', 'adicao', 'adição', 'soma']:
26         resultado = adicao(num1, num2)
27     elif operacao in ['-', 'subtracao', 'subtração', 'menos']:
28         resultado = subtracao(num1, num2)
29     elif operacao in ['*', 'multiplicacao', 'multiplicação', 'x', 'vezes']:
30         resultado = multiplicacao(num1, num2)
31     elif operacao in ['/', 'divisao', 'divisão']:
32         resultado = divisao(num1, num2)
33     else:
34         resultado = 'Operação inválida'
35
36     return resultado
37
38 #Laço Repetição
39 while saida.lower() != 'n':
40     try:
41         num1 = float(input('Digite o primeiro número: '))
42         num2 = float(input('Digite o segundo número: '))
43         operacao = input('Digite a operação desejada (+, -, *, / ou nome): ')
44
45         resultado = calculadora(num1, num2, operacao)
46         print(f'Resultado da operação: {resultado}')
47     except ValueError:
48         print('Entrada inválida! Por favor, digite apenas números')
49
50 saida = input("Deseja continuar? (S/N): ")

```

```
PS D:\Cursos\Faculdade\Microatividades> & python d:/Cursos/Faculdade/Microatividades/calculadora_v2.py
Digite o primeiro número: 1
Digite o segundo número: 3
Digite a operação desejada (+, -, *, / ou nome): +
Resultado da operação: 4.0
Deseja continuar? (S/N): 2
Digite o primeiro número: 2
Digite o segundo número: 2
Digite a operação desejada (+, -, *, / ou nome): -
Resultado da operação: 0.0
Deseja continuar? (S/N): s
Digite o primeiro número: 6
Digite o segundo número: 4
Digite a operação desejada (+, -, *, / ou nome): *
Resultado da operação: 24.0
Deseja continuar? (S/N): s
Digite o primeiro número: 4
Digite o segundo número: 0
Digite a operação desejada (+, -, *, / ou nome): /
Não foi possível realizar a divisão por 0
Resultado da operação: None
Deseja continuar? (S/N): s
Digite o primeiro número: 6
Digite o segundo número: 2
Digite a operação desejada (+, -, *, / ou nome): /
Resultado da operação: 3.0
Deseja continuar? (S/N): █
```