Projetos em Python

Este repositório contém exemplos de códigos em Python, demonstrando o uso de estruturas condicionais, de repetição e funções, além de implementações de uma calculadora simples.

Exemplo de Código: Estruturas de Condição

Nome do arquivo: `estruturas_condicao1.py`

Descrição:

Este script simples verifica a temperatura e imprime uma mensagem com base no valor fornecido. Se a temperatura for inferior a 30 graus, ele informa que está amena, caso contrário, indica que está calor.

Código:

```python
temperatura = 31

if temperatura < 30:
   print('A temperatura hoje está amena')
else:
   print('Hoje está fazendo calor')
```

Como usar:

Salve o código acima em um arquivo chamado estruturas_condicao1.py.

Execute o script com Python:

```bash
python estruturas_condicao1.py
```

```

O resultado será exibido no terminal, indicando se a temperatura está amena ou quente, com base no valor atribuído à variável temperatura.

## Exemplo de Código: Estruturas de Condição

**Nome do arquivo:** `estruturas_condicao2.py`

### Descrição:

Este script avalia o tempo de experiência de uma pessoa e classifica seu nível de conhecimento em três categorias: junior, pleno ou senior, com base nos anos de experiência fornecidos.

### Código:

```python

tempoExperiencia = 3

if tempoExperiencia < 2:
  print('Nivel de conhecimento junior.')
elif tempoExperiencia > 2 and tempoExperiencia < 5:
  print('Nivel de conhecimento pleno.')
else:
  print('Nivel de conhecimento senior.')
```

Como usar:

Salve o código acima em um arquivo chamado estruturas_condicao2.py.

Execute o script com Python:
```

```bash
python estruturas_condicao2.py
```

O resultado exibirá no terminal o nível de conhecimento de acordo com o valor definido para a variável tempoExperiencia.

Exemplo de Código: Estruturas de Repetição

Nome do arquivo: estruturas_repeticao1.py

Descrição:

Este script solicita ao usuário a entrada de um número indefinidamente até que o valor 0 seja inserido. O programa também valida se a entrada é um número válido, exibindo uma mensagem correspondente.

Código:

```python
entrada_idade = None
while entrada_idade != 0:
    entrada_idade = input('Digite um número qualquer ou 0 para sair: ')
    if entrada_idade.isdigit():
        entrada_idade = int(entrada_idade)
        print(f'Número digitado: {entrada_idade}')
    else:
        print('Digite um número válido!')
```

Como usar:

Salve o código acima em um arquivo chamado estruturas_repeticao1.py.

Execute o script com Python:

```bash
python estruturas_repeticao1.py
```

O programa continuará solicitando números ao usuário e exibindo os valores digitados até que o número 0 seja inserido, momento em que o loop será encerrado.

Exemplo de Código: Estruturas de Repetição

Nome do arquivo: estruturas_repeticao2.py

Descrição:

Este script demonstra o uso de loops for em duas situações. No primeiro loop, percorre cada caractere de uma string e imprime cada um deles. No segundo loop, percorre um intervalo de números de 1 a 10 e imprime cada número.

Código:

```python
texto = 'Ola, laco for.'
for item in texto:
  print(f'Caractere: {item}')

for item in range(1, 11):
  print(f'Numero do intervalo: {str(item)}')
```

Como usar:

Salve o código acima em um arquivo chamado estruturas_repeticao2.py.

Execute o script com Python:

```bash
python estruturas_repeticao2.py
```

O primeiro loop irá imprimir cada caractere da string "Ola, laco for.". O segundo loop irá imprimir os números de 1 a 10.

Exemplo de Código: Funções

Nome do arquivo: funcoes1.py

Descrição:

Este script demonstra a criação e utilização de uma função simples chamada imprimir_variavel, que define uma variável texto dentro da função e imprime o conteúdo da variável.

Código:

```python
def imprimir_variavel():
    texto = 'Ola, funcoes em Python'
    print(texto)

imprimir_variavel()
```

Como usar:

Salve o código acima em um arquivo chamado funcoes1.py.

Execute o script com Python:

```bash
python funcoes1.py
```

O resultado será a impressão da mensagem "Ola, funcoes em Python" no terminal.

Exemplo de Código: Funções

Nome do arquivo: funcoes2.py

Descrição:

Este script define uma função chamada loginUsuario, que recebe um parâmetro perfil e exibe uma mensagem de boas-vindas de acordo com o perfil do usuário. Se o perfil for "admin" (não importa se em maiúsculas ou minúsculas), o programa exibe uma saudação de administrador, caso contrário, exibe uma saudação genérica para usuários.

Código:

```python
def loginUsuario(perfil):
  if perfil.lower() == 'admin':
    print('Bem-vindo, Administrador')
  else:
    print('Bem-vindo, Usuário')

loginUsuario('Admin')
```

loginUsuario('admin')

loginUsuario('User')

loginUsuario('usuario')

loginUsuario('etc')

```

## Como usar:

Salve o código acima em um arquivo chamado funcoes2.py.

Execute o script com Python:

```bash

python funcoes2.py

```

O programa executará a função loginUsuario com diferentes perfis e exibirá mensagens de boas-vindas dependendo do valor inserido:

- Para "Admin" ou "admin", a mensagem será "Bem-vindo, Administrador".
- Para qualquer outro valor, a mensagem será "Bem-vindo, Usuário".

Exemplo de Código: Calculadora

Nome do arquivo: calculadora_v1.py

Descrição:

Este script implementa uma calculadora simples em Python, que permite ao usuário realizar operações básicas, como soma, subtração, multiplicação, divisão, potenciação, raiz e resto da divisão. O usuário pode inserir os números e a operação desejada, e o resultado será exibido até que o comando "sair" seja inserido.
```

Código:

```python

def soma(a, b):
    return a + b

def subtracao(a, b):
    return a - b

def multiplicacao(a, b):
    return a * b

def divisao(a, b):
    return a / b

def potencia(a, b):
    return a ** b

def raiz(a, b):
    return a ** (1/b)

def resto(a, b):
    return a % b

operacoes = {
    '+': soma,
    '-': subtracao,
    '*': multiplicacao,
    '/': divisao,
    '**': potencia,
    '//': raiz,
```

```
    '%': resto
}

while True:
    a = float(input('Digite um número: '))
    op = input('Digite uma operação (ou "sair" para sair): ')
    if op.lower() == 'sair':
        break
    b = float(input('Digite outro número: '))

    if op not in operacoes:
        print('Operação inválida!')
        continue

    resultado = operacoes[op](a, b)
    print(f'{a} {op} {b} = {resultado}')
```

Como usar:

Salve o código acima em um arquivo chamado calculadora_v1.py.

Execute o script com Python:

```bash
python calculadora_v1.py
```

Siga as instruções exibidas no terminal:

- Digite um número.
- Escolha uma operação entre as disponíveis (+, -, *, /, **, //, %).
- Digite outro número.

- O resultado da operação será exibido.

- Para encerrar o programa, digite "sair".

Exemplo de Código: Calculadora Versão 2

Nome do arquivo: calculadora_v2.py

Descrição:

Este script implementa uma versão melhorada de uma calculadora simples em Python. Ele permite que o usuário realize operações básicas, como adição, subtração, multiplicação e divisão, com tratamento para divisão por zero. O usuário pode continuar fazendo cálculos até que escolha sair.

Código:

```python

saida = ''

def adicao(a, b):
  return a + b

def subtracao(a, b):
  return a - b

def multiplicacao(a, b):
  return a * b

def divisao(a, b):
  if b == 0:
    return "Não foi possível realizar a divisão por 0"
  else:
```

```
    return a / b

def calculadora(a, b, op):
  if op == '+':
    return adicao(a, b)
  elif op == '-':
    return subtracao(a, b)
  elif op == '*':
    return multiplicacao(a, b)
  elif op == '/':
    return divisao(a, b)

while saida != 'n':
  num1 = float(input("Digite o primeiro número: "))
  num2 = float(input("Digite o segundo número: "))
  op = input("Digite a operação (+, -, *, /): ")
  resultado = calculadora(num1, num2, op)
  print(f"Resultado da operação: {resultado}")
  saida = input("Deseja continuar? (S/N): ").lower()
```

Como usar:

Salve o código acima em um arquivo chamado calculadora_v2.py.

Execute o script com Python:

```bash
python calculadora_v2.py
```

Siga as instruções exibidas no terminal:

- Digite o primeiro número.
- Digite o segundo número.
- Escolha uma operação entre as disponíveis (+, -, *, /).
- O resultado da operação será exibido.
- Para realizar outra operação, digite "S". Para sair, digite "N".

Licença

--

Contato e Suporte

Para dúvidas ou suporte, entre em contato através do e-mail: drtizoco@gmail.com

Recursos Adicionais

- [Documentação do Python](https://docs.python.org/3/)
- [Tutorial de Python](https://docs.python.org/3/tutorial/index.html)