

Computational Thinking Using Python

FIAP

FIAP

COMPUTATIONAL THINKING USING PYTHON

Prof. Edson de Oliveira

Computational Thinking Using Python

Entrada | Saída | Processamento de dados

variável

Entrada de dados — Interação do usuário como algoritmo (teclado)
Python: `input()`

Mensagem

Saída de dados — Interação do Algoritmo como usuário (Monitor)
Python: `print()`

Var = calculo

Processamento de dados — Cálculos (CPU)
Python: `<cálculo>`

Computational Thinking Using Python

Exemplo de construção de Algoritmo

Problema: Fazer um algoritmo que pegue dois valores do usuário e calcule a média

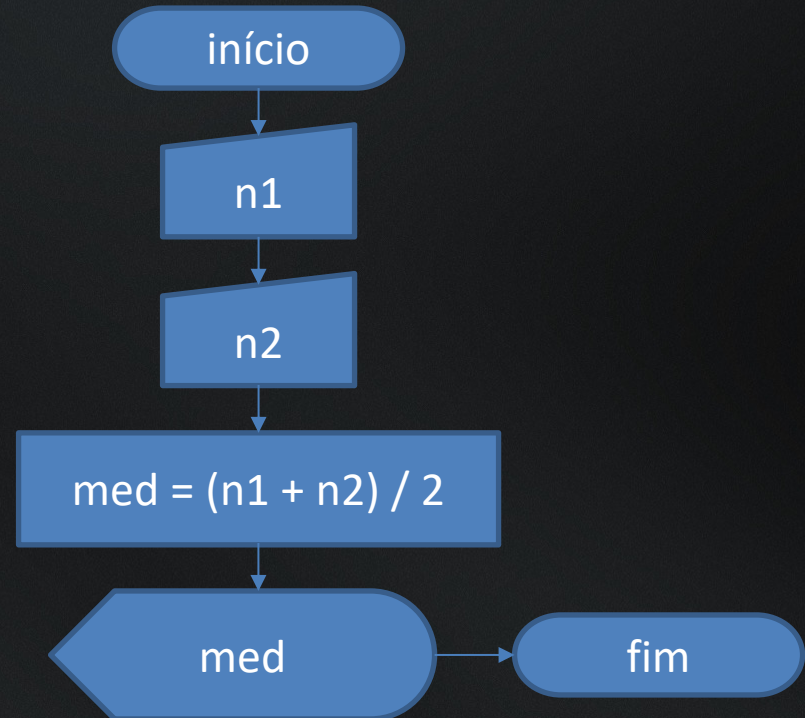
NARRATIVA (os passos do algoritmo):

- Digitar o primeiro numero (ed)
- Digitar o segundo numero (ed)
- Calcular a média dos 2 números (pd)
- Exibir o resultado (sd)

Tela: Execução do algoritmo

```
20
10
15
```

FLUXOGRAMA: Desenho (ideia) do Algoritmo



Computational Thinking Using Python

Algoritmo Completo

1. Dada a **quilometragem** parcial de um carro e a **quantidade** de litros gastos ele para percorrer esta quilometragem, fazer um algoritmo que calcule quantos **Km/l** o carro percorreu.

Entrada: 345.6 | 25.4 Saída: 13.6

Entrada: 556.1 | 59.7 Saída: 9.31

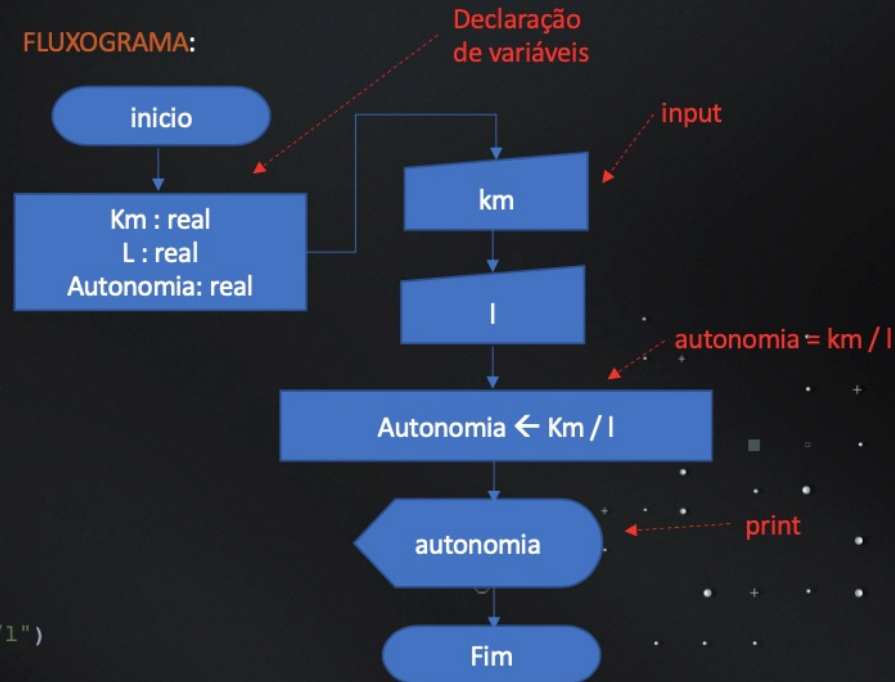
NARRATIVA:

- Ler a quilometragem (Entrada de dados)
- Ler A quantidade de litros gastos
- Calcular a autonomia (proc. Dados)
- Exibir a autonomia (saída de dados)

CÓDIGO EM PYTHON:

```
# Ler a quilometragem (Entrada de dados)
km = input("Km percorrida: ")
km = float(km)
# Ler A quantidade de litros gastos
l = input("Litros gastos: ")
l = float(l)
# Calcular a autonomia (proc. Dados)
autonomia = km / l
# Exibir a autonomia (saída de dados)
print(f"Autonomia: {autonomia:10.2f} km/l")
```

FLUXOGRAMA:



Computational Thinking Using Python

Algoritmo Completo

2. Dado o **preço** do maço de cigarros, a **quantidade** de maços consumidos por dia e o **tempo** em anos que a pessoa fuma, calcular o quanto esta pessoa já **gastou** fumando.

Entrada: 10 | 1 | 3 Saída: 10950

Entrada: 11.5 | 2 | 5 Saída: 41975

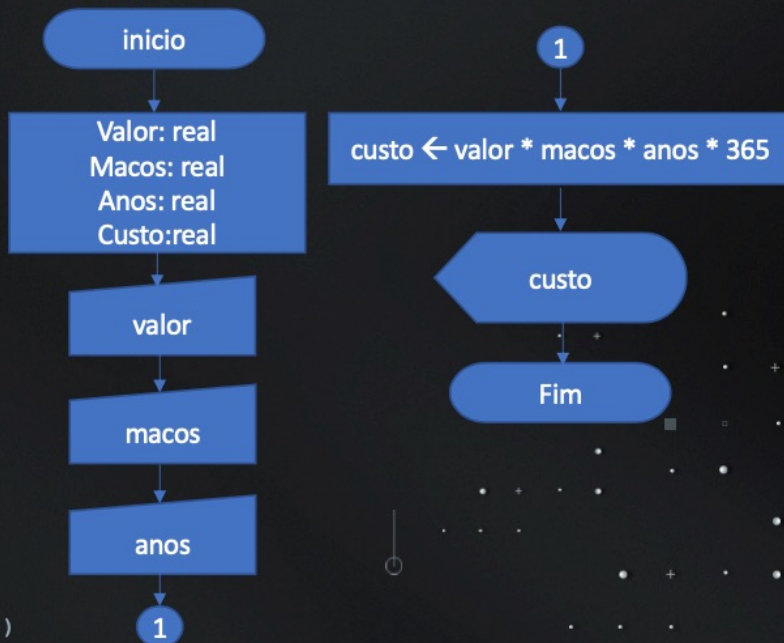
NARRATIVA:

- Ler o valor do maço
- Ler a qtd de macos fumados por dia
- Ler o tempo em anos
- Calcular o quanto gastou fumando
- Exibir o valor gasto

CÓDIGO EM PYTHON:

```
# Ler o valor do maço
valor = input("Digite o valor do maço: R$ ")
valor = float(valor)
# Ler a qtd de macos fumados por dia
macos = input("Qtd macos por dia: ")
macos = float(macos)
# Ler o tempo em anos
anos = input("Qtd anos: ")
anos = float(anos)
# Calcular o quanto gastou fumando
custo = valor * macos * anos * 365
# Exibir o valor gasto
print(f"Você já gastou R$ {custo:.2f} fumando!")
```

FLUXOGRAMA:



Computational Thinking Using Python

Algoritmo Completo

3. Um caixa eletrônico dispensa cédulas de 50, 20 e 10 reais. Considerando que a **quantia** seja múltiplo de 10, fazer um algoritmo que exiba um relatório com **quantas cédulas de cada cédula** são necessárias para compor esta quantia.

Entrada: 130 Saída: 50=2 | 20=1 | 10=1

Entrada: 270 Saída: 50=5 | 20=1 | 10=0

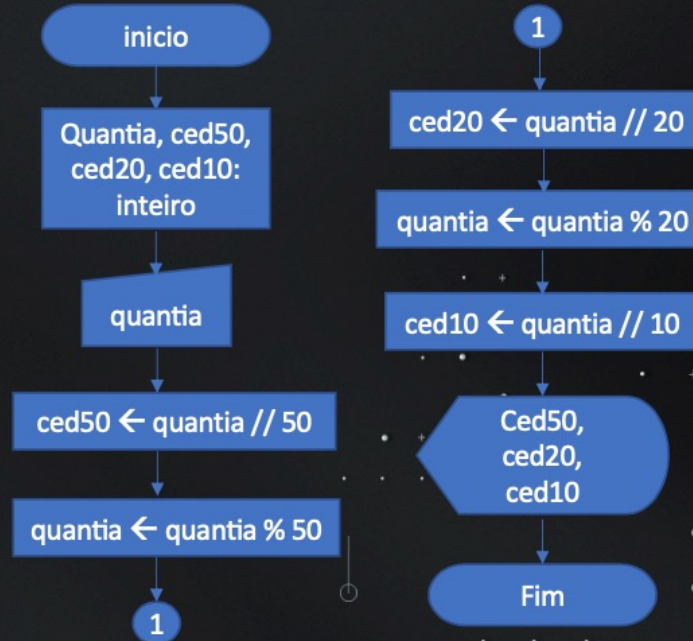
NARRATIVA:

- Ler a quantia a ser sacada
- Descobrir quantas cédulas de 50
- Descobrir quantas cédulas de 20
- Descobrir quantas cédulas de 10
- Exibir as quantidades de cada cédula

CÓDIGO EM PYTHON:

```
# Ler a quantia a ser sacada
quantia = input("Digite a quantia a ser sacada: ")
quantia = int(quantia)
# Descobrir quantas cédulas de 50
ced50 = quantia // 50
quantia = quantia % 50
# Descobrir quantas cédulas de 20
ced20 = quantia // 20
quantia = quantia % 20
# Descobrir quantas cédulas de 10
ced10 = quantia // 10
# Exibir as quantidades de cada cédula
print(f"R$50,00={ced50}\nR$20,00={ced20}\nR$10,00={ced10}\n")
```

FLUXOGRAMA:



Computational Thinking Using Python

Casting – Mudança de tipo da variável

TIPO DO DADO

Inteiro

Real

Texto

Lógico

REPRESENTAÇÃO PYTHON()

int()

float()

str()

bool()

EXEMPLOS

```
# Valor original – str()
valor = "45"
print(f"Conteudo = {valor} \t| Tipo = {type(valor)}")
# casting int()
valor = int(valor)
print(f"Conteudo = {valor} \t| Tipo = {type(valor)}")
# casting float()
valor = float(valor)
print(f"Conteudo = {valor} | Tipo = {type(valor)}")
# casting str()
valor = str(valor)
print(f"Conteudo = {valor} | Tipo = {type(valor)}")
# casting bool()
valor = bool(valor)
print(f"Conteudo = {valor} | Tipo = {type(valor)}")
```

SAÍDA

```
Conteudo = 45      | Tipo = <class 'str'>
Conteudo = 45      | Tipo = <class 'int'>
Conteudo = 45.0    | Tipo = <class 'float'>
Conteudo = 45.0    | Tipo = <class 'str'>
Conteudo = True    | Tipo = <class 'bool'>
```


Computational Thinking Using Python

Formatações – print()

Código-fonte

```
# FORMATAÇÕES DO PRINT
nome = "Edson"
idade = 48
salario = 34537.94745345
print("----- Forma 1 - Separando com vírgulas")
print(nome, idade, salario)
print("----- Forma 2 - Separando com vírgulas")
print("Nome: ", nome, "Idade: ", idade, "Salario:", salario)
print("----- Forma 3 - Separando com vírgulas | quebrando a linha")
print("Nome: ", nome, "\nIdade: ", idade, "\nSalario:", salario) # 0 \n força a mudança de linha
```

Execução

```
----- Forma 1 - Separando com vírgulas
Edson 48 34537.94745345
----- Forma 2 - Separando com vírgulas
Nome:  Edson Idade:  48 Salario: 34537.94745345
----- Forma 3 - Separando com vírgulas | quebrando a linha
Nome:  Edson
Idade:  48
Salario: 34537.94745345
```

Computational Thinking Using Python

Formatações – print()

Código-fonte

```
# Forma 4 - Utilizando o format()
print("Nome: {} Idade: {} Salario: {}".format(nome, idade, salario))
# Forma 5
print("Nome: {} Idade: {} Salario: {} - Obrigado {}".format(nome, idade, salario))
# Forma 6
print("Nome: {n} Idade: {i} Salario: {s} ".format(n = nome, i = idade, s = salario))
# Forma 7 - utilizando o print'f'
print(f"Nome: {nome} Idade: {idade} Salario: {salario}")
```

Execução

```
Nome: Edson Idade: 48 Salario: 34537.94745345
Nome: Edson Idade: 48 Salario: 34537.94745345 - Obrigado Edson
Nome: Edson Idade: 48 Salario: 34537.94745345
Nome: Edson Idade: 48 Salario: 34537.94745345
```

Computational Thinking Using Python

Formatações – print()

Código-fonte

```
# Forma 8 - utilizando o print'f'
idade = 8
print(f"Nome.....: {nome} \nIdade.....: {idade:05d} \nSalario.....: {salario:.2f}")
valor1 = 45.93838
valor2 = 234.1
valor3 = 234244.0
valor4 = 2
print("Extrato:")
print(f"Valor 1: R$ {valor1:10.2f}")
print(f"Valor 2: R$ {valor2:10.2f}")
print(f"Valor 3: R$ {valor3:10.2f}")
print(f"Valor 4: R$ {valor4:10.2f}")
```

Execução

```
Nome.....: Edson
Idade.....: 00008
Salario.....: 34537.95
Extrato:
Valor 1: R$      45.94
Valor 2: R$     234.10
Valor 3: R$  234244.00
Valor 4: R$      2.00
```


Computational Thinking Using Python

Formatações – print()

Código-fonte

```
valor1 = 45.93838
valor2 = 234.1
valor3 = 234244.0
valor4 = 2
print(f"""
    Extrato:
    R$ {valor1:10.2f}
    R$ {valor2:10.2f}
    R$ {valor3:10.2f}
    R$ {valor4:10.2f}
""")
```

Execução

```
Extrato:
Valor 1: R$      45.94
Valor 2: R$     234.10
Valor 3: R$  234244.00
Valor 4: R$       2.00
```

Computational Thinking Using Python

Lista de exercícios não avaliativos

1. Fazer um algoritmo que peça um numero ao usuário e calcule o seu quadrado.

Entrada: 4

Saída: 16

2. Fazer um algoritmo que calcule o dobro de um número dado pelo usuário.

Entrada: 20

Saída: 40

3. Fazer um algoritmo que calcule a média de 4 números dados pelo usuário.

Entrada: 4 6 5 7

Saída: 5.5

4. Dada a quilometragem parcial de um carro (km) e a quantidade de litros (l) gastos para percorrer esta quilometragem, fazer um algoritmo que calcule quantos Km/l o carro consumiu.

Entrada: 452 41

Saída: 11.02

5. Dado o preço do maço de cigarros, a quantidade de maços consumidos por dia e o tempo em anos que a pessoa fuma, calcular quanto esta pessoa gastou com cigarros.

Entrada: 10 1 2

Saída: 7300

6. Um caixa eletrônico dispensa cédulas de 50, 20 e 10 reais. Considerando que a quantia já seja múltipla de 10, fazer um algoritmo que exiba um relatório com quantas cédulas de cada são necessárias para compor a quantia.

Entrada: 130

Saída: 50: 2 20: 1 10: 1

FIAP

COMPUTATIONAL THINKING USING PYTHON

Edson de Oliveira