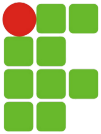

# Fundamentos de Programação

Diego Silveira Costa Nascimento

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
diego.nascimento@ifrn.edu.br

7 de setembro de 2022

# Ementa

1. Introdução
2. Saída
3. Entrada
4. Seleção
5. Repetição
6. Modularização

# Ementa do Curso



# Python

**Definição**

É uma linguagem de script de propósito geral, podendo ser usada para criar qualquer tipo de software.

- Foi concebido no final de 1989 por Guido van Rossum; e
- O nome Python teve a sua origem no grupo humorístico britânico Monty Python.

# Características

- É uma linguagem interpretada;
- Os tipos das variáveis são determinados dinamicamente;
- Oferece tipos de alto nível;
- É orientada a objetos; e
- É multi-plataforma.

# Programa em Python

- Um programa em Python pode ser escrito em qualquer editor de texto;
- O documento com o código fonte deve ser salvo com extensão .py;
- Para facilitar o desenvolvimento é comum utilizar-se um IDE (Integrated Development Environment); e
- O IDLE é o ambiente de desenvelvimento padrão.

# Ementa do Curso



# Saída de Dados

## Definição

A instrução de saída de dados é a instrução através da qual o computador se comunica com usuário durante a execução do programa. Isso é feito, geralmente, através da exibição de alguma informação na tela.

- Em Python existe apenas um comando para instrução de saída: print.

## Exemplo

```
print('Oi, mundo!')
```

# Comentário

## Definição

É uma estrutura da linguagem que permite ao desenvolvedor fazer uma breve explicação do código escrito.

- Comentários são iniciados com #.

## Exemplo

```
# Descrição da funcionalidade desenvolvida
# Autor : Diego
print ('Testando!')
```

## Importante

O que for escrito no bloco de comentário será ignorado pelo interpretador.

# Ementa do Curso



# Variável

- Uma variável representa uma posição de memória;
- Possui um nome e tipo;
- Seu conteúdo pode variar ao longo do tempo, durante a execução do programa;
- Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante;
- Não existe limite para o número de variáveis em um programa; e
- Cada variável criada ocupa um espaço de memória de acordo com seu tipo e seu tamanho.

# Declaração de Variável

- A tipagem de Python é dinâmica;
- Logo não é necessário declarar os tipos de variáveis;
- Devem ser declaradas inicialmente por letras ($a - z, A - Z$) ou sublinhado (_);
- Acentuação é permitida (não é recomendado); e
- É case sensitive ($a \neq A$).

## Exemplos

```
i
nome
data_nascimento
nota1
_sexo
mediaGeral
```

# Tipos de Variável

- str : Cadeia de caracteres;
- int : Inteiro;
- float : Ponto flutuante ou real; e
- bool : Lógico ou booleano.

## Exemplos

```
type('Python')
type(36)
type(82.5)
type(True)
```

# Operador de Atribuição

## Definição

O comando de atribuição é utilizado para conceder valores ou operações a variáveis.

- Em python o operador de atribuição é o sinal de igual: $=$;
- Do lado esquerdo ao operador de atribuição fica a variável à qual está sendo atribuído o valor; e
- A direita do operador pode-se escrever qualquer expressão (constantes, variáveis ou expressões numéricas).

## Exemplos

```
linguagem = 'Python'
idade = 36
altura = 82.5
matriculado = True
```

# Entrada de Dados

## Definição

É o meio pelos quais os dados são transferidas pelo usuário ou pelos níveis secundários de memória ao computador.

- Python possui o comando para instrução de entrada via teclado: input().

## Exemplo

```
nome = input('Digite seu nome:')
print('Oi,',nome)
```

# Operadores Aritméticos

## Definição

A aritmética é o ramo da matemática que lida com números e com as operações possíveis entre eles.

- + : Adição;
- - : Subtração;
- * : Multiplicação;
- / : Divisão;
- // : Divisão inteira;
- % : Resto; e
- ** : Potência.

## Exemplo

```
a = int(input('Digite um nú mero inteiro:'))
b = int(input('Digite um nú mero inteiro:'))
c = a + b
print ('Resultado =',c)
```

# Expressão Aritmética

## Definição

Uma expressão constitui-se em um conjunto de variáveis e/ou valores, separados por caracteres especiais, que indicam as operações que devem ser executadas.

## Importante

Os operadores devem obedecer uma ordem de precedência:

- Parênteses;
- Potenciação;
- Multiplicação, Divisão e Resto; e
- Adição e subtração.

## Exemplo

```
a = 2
b = 8
c = a + b / 2
print(c)
```

# Ementa do Curso



# Estrutura de Seleção

## Definição

Também citado na literatura por Estrutura Condicional, é a representação de um ou mais comandos de decisão que são responsáveis por mudar o fluxo das instruções de um algoritmo em tempo de execução.

- Python possui apenas uma estrutura de controle: if

## Exemplo

```
status = 0
if status == 0:
   print('Livre')
else:
   print('Ocupado')
```

## Importante

O comando else não é obrigatório.

# Operadores Relacionais

## Definição

Os operadores relacionais estabelecem uma relação entre seus operandos.

- $==$ : igual;
- $!=$ : diferente;
- $<$ : menor;
- $<=$ : menor ou igual;
- $>$ : maior; e
- $>=$ : maior ou igual.

## Exemplo

```
numero = int(input('Digite um número:'))
if numero >= 0:
    print('Número positivo')
else:
    print('Número negativo')
```

# Operadores Lógicos

## Definição

Os operadores lógicos definem as maneiras como as relações podem ser conectadas.

- not : negação lógica;
- and : e lógico; e
- or : ou lógico.

## Exemplo

```
nota = float(input('Digite uma nota:'))
if (nota >= 0) and (nota < 11):
    print('Nota válida')
else:
    print('Nota inválida')
```

# Tabela-verdade

## Exemplo

| a = | b = | a and b | a or b | not a |
|---|---|---|---|---|
| True | True | True | True | False |
| True | False | False | True | False |
| False | True | False | True | True |
| False | False | False | False | True |

# Estrutura de Seleção Aninhada

## Definição

É uma estrutura para desvio de fluxo do programa formada pelo comando de decisão if / elif / else mais subestruturas de decisão.

## Exemplo

```
numero = int(input('Digite um número inteiro:'))
if numero > 0:
    print ('Número positivo')
elif numero < 0:
    print ('Número negativo')
else:
    print ('O número digitado foi zero')
```

# Ementa do Curso



# Estruturas de Repetição

**Definição**

Uma estrutura de repetição é uma estrutura de desvio do fluxo de controle presente em linguagens de programação que realiza e repete diferentes computações ou ações.

- Python possui duas estruturas de repetição:
  - while; e
  - for.

# Estrutura While

## Definição

A construção while (também chamada repetição pré-testada) é a mais difundida estrutura de repetição.

## Exemplo

```
i = 1
while i <= 10:
    print (i)
    i = i + 1
```

# Comando Break

## Definição

O comando break permite parar uma execução de uma instrução de repetição toda vez que o mesmo for invocado, ignorando, caso ainda existam, outras instruções a serem executadas.

## Exemplo

```
i = 1
while i <= 10:
    print (i)
    if i == 5:
        break
    i = i + 1
```

# Estrutura For

## Definição

A construção for, ou repetição com variável de controle, é uma estrutura de repetição que designa uma variável de controle para cada iteração do bloco, e uma operação de passo a cada iteração.

## Exemplo

```
for i in range(11):
    print(i)
```

# Ementa do Curso



# Função

## Definição

São subrotinas (módulos ou métodos) de programas, capazes de executar uma tarefa definida pelo programador, que pode retorna ou não algum valor. Os programas desenvolvidos com subprogramas são ditos modulares.

- Python possui uma estrutura para definição de função: def.

# Função

## Exemplo 1

```
def mensagem():
    print('Oi , mundo!')

mensagem()
```

## Exemplo 2

```
def somar(a,b):
    return a + b

valor1 = float(input('Digite o primeiro valor :'))
valor2 = float(input('Digite o segundo valor :'))
resultado = somar(valor1,valor2)
print('A soma de ',valor1 ,' + ',valor2 ,' = ',resultado)
```

## Definição

É quando uma função refere-se a si própria durante a própria definição.

## Exemplo 2

```
def contador(i):
    if i > 1:
        contador(i - 1)
    print (i)

contador(10)
```