fepi
Centro Universitário de Itajubá

CENTRO UNIVERSITÁRIO DE ITAJUBÁ

Python

Variáveis

Inteiros, Float
Booleanos e strings

PYTHON – INTEIROS, FLOAT, BOOLEANOS, STRINGS

aula1.py > ...

```python
# Inteiros
idade = 30
saldo_bancario = -2900

# Floats
preco_unitario = 4.99
taxa_de_juros = 9.025

# Booleanos
verdadeiro = True
falso = False

# Strings
nome = "Maria"
endereco = "Rua das Flores, 123"  # Corrigido para uma string completa

# Exibindo as variáveis
print(idade)
print(taxa_de_juros)
print(verdadeiro)
print(endereco)
```

Listas, Tuplas
Conjuntos e Dicionários

PYTHON – LISTAS, TUPLAS, CONJUNTOS, DICIONÁRIOS

aula2.py > ...

```python
# listas
numeros = [1, 2, 3, 4, 5]
nomes = ["João", "Maria", "José"]
misturado = [1,"dois", True]
# tuplas
coordenadas =(10, 20)
cores = ("vermelho" ,"verde", "azul")
# conjuntos
vogais = {"a", "e", "i", "o", "u"}
# dicionários
pessoa = {"nome": "João" ,"idade": 39,"profissão":"engenheiro"}
notas = {"matematica": 8.5, "portugues": 7.0, "historia": 9.5}
print(numeros)
print(misturado)
print(coordenadas)
print(vogais)
print(pessoa)
```

Operadores

PYTHON - OPERADORES

aula3.py > ...

```python
a = 10
b = 3
print("Resultados")
# adição
resultado = a + b
print(resultado) # saída: 13
# subtração
resultado = a - b
print(resultado) # saída: 7
# multiplicação
resultado = a * b
print(resultado) # saída: 30
# divisão
resultado = a / b
print(resultado) # saída: 3.33333333333
# divisão inteira
resultado = a // b
print(resultado) # saída: 3
# resto
resultado = a % b
print(resultado) # saída: 1
# potência
resultado = a ** b
print(resultado) # saída: 1000
```

Indentação

Indentação

Em Python, a indentação possui função bastante especial, até porque, os blocos de instrução são delimitados pela profundidade da indentação, isto é, os códigos que estiverem rente a margem esquerda, farão parte do primeiro nível hierárquico. Já, os códigos que estiverem a 4 espaços da margem esquerda, estarão no segundo nível hierárquico e aqueles que estiverem a 8 espaços, estarão no terceiro nível e assim por diante.

Os blocos são uma ou mais instruções que devem ser executadas uma após a outra, de cima para baixo da esquerda para a direita.

PYTHON – EXEMPLOS DE INDENTAÇÃO

```python
nome = "Aluno"

if nome:
    print(f"Olá, {nome}!")
else:
    print("Olá, mundo!")
```

```python
contador = 0

while contador < 5:
    print(f"Contador vale: {contador}")
    contador += 1
```

Estruturas de Controle

PYTHON - APLICAÇÕES

Declaração if: permite executar um bloco de código somente se uma condição especificada for verdadeira.

Equivalente no VisualG: Se () então .... fimse

aula4.py > ...

```
1   a = 8
2   b = 7
3   if a > b:
4       print("a maior que b")
5   if a < b:
6       print("a menor que b")
7   if a == b:
8       print("a e igual a b")
```

PYTHON - APLICAÇÕES

Declaração if : else: permite executar um bloco de código somente se uma condição especificada for verdadeira e outra parte do código se for falsa.

Equivalente no VisualG: Se () então .. senao ... fimse

aula5.py > ...

```
1   idade = 18
2   if idade >= 18:
3       print("Você é maior de idade")
4   else:
5       print("Você é menor de idade")
```

Estruturas de Repetição

PYTHON - FOR

Declaração for: permite iterar sobre uma sequência de elementos.

Equivalente no VisualG: Se

aula6.py > ...

```python
numeros = [1, 2, 3, 4, 5]
for numero in numeros:
    print(numero)
```

```
PS D:\0-Fepi\Ano-2025-1º Semestre\Informatica\pytho
1
2
3
4
5
```

PYTHON - WHILE

Declaração while: permite executar um bloco de código enquanto uma condição especificada for verdadeira.

Equivalente no VisualG: enquanto faca ..

aula7.py > ...

```
1   contador = 0
2   while contador < 5:
3       print("o contador é",contador)
4       contador += 1
```

```
PS D:\0-Fepi\Ano-2024-1º Semestre\Back End\Python> python .\teste7.py
O contador é 0
O contador é 1
O contador é 2
O contador é 3
O contador é 4
```

Declaração break

PYTHON – DECLARAÇÃO BLEAK

Declaração break: permite sair de um loop antecipadamente.

aula8.py > ...

```
1    numeros = [1, 2, 3, 4, 5]
2
3    for numero in numeros:
4        if numero == 3:
5            break
6        print(numero)
```

```
PS D:\0-Fepi\Ano-2025-1º Semestre\Infor
1
2
```

Declaração continue

PYTHON – DECLARAÇÃO CONTINUE

Declaração continue: permite continuar para a próxima iteração de um loop.

aula9.py > ...

```
1   numeros = [1, 2, 3, 4, 5]
2
3   for numero in numeros:
4       if numero == 3:
5           continue
6       print(numero)
```

```
PS D:\0-Fepi\Ano-2025-1º Semestre\Infor
1
2
4
5
```

try/except

PYTHON - TRY/EXCEPT

A estrutura try/except em Python é usada para tratar erros (também chamados de exceções) que podem ocorrer durante a execução do código. Em vez de o programa parar com um erro, você pode capturar esse erro e reagir de forma controlada.

aula10.py > ...

```
1   try :
2       resultado = 16/0
3   except ZeroDivisionError:
4       print("Erro: divisão por zero")
```

Funções

PYTHON - FUNÇÕES

Em Python, uma função é um bloco de código que executa uma determinada tarefa. As funções são usadas para modularizar o código, tornando-o mais organizado e fácil de entender e manter. Para definir uma função em Python, usamos a palavra-chave def, seguida pelo nome da função e seus parâmetros (se houver), seguida pelo corpo da função.

PYTHON – FUNÇÕES EXEMPLO

aula11.py > ...

```python
def media(numeros) :
    soma = 0
    for numero in numeros:
        soma += numero
    return soma / len(numeros)

# maneira de chamar a função
numeros = [1, 2, 3, 4, 5]
resultado = media(numeros)
print(resultado) # saída: 3
```

Função input

PYTHON – FUNÇÃO INPUT

A função input() do Python serve para receber dados do usuário via teclado enquanto o programa está rodando.

```
aula12.py > ...
1    nome = input("Digite seu nome: ")
2    print(f"Olá, {nome}!")
```

- ✓ O texto dentro do input() (por exemplo, "Digite seu nome: ") é exibido na tela como mensagem para o usuário.
- ✓ O programa espera o usuário digitar algo e apertar Enter.
- ✓ O valor digitado é sempre retornado como string (texto), mesmo que a pessoa digite um número, e atribuído a variável nome.

PYTHON - APLICAÇÕES

Para usar como número:

aula13.py > ...

```
1   idade = int(input("Digite sua idade: "))
2   print(idade + 5)
```

Função print

PYTHON - APLICAÇÕES

A função print() no Python é usada para mostrar informações na tela — ou seja, exibir textos, números, resultados de cálculos, variáveis, etc.

aula14.py

```
1    print("Olá, mundo!")
```

Imprimindo variáveis:

aula15.py > ...

```
1    nome = "Ana"
2    print("Seu nome é", nome)
```

PYTHON - APLICAÇÕES

Com f-string (forma moderna):

aula16.py > ...

```
1    idade = 25
2    print(f"Você tem {idade} anos.")
```

Vários valores:

aula17.py

```
1    print("A soma de", 3, "+", 2, "é", 3 + 2)
```

PYTHON - APLICAÇÕES

Cada print() pula uma linha automaticamente.

Se quiser manter na mesma linha:

aula18.py

```
1   # Cada print pula linha
2   print("Olá")
3   print("mundo!")
4   # Para não pular a linha
5   print("Olá", end=" ")
6   print("mundo!")
```

fepi
Centro Universitário de Itajubá

CENTRO UNIVERSITÁRIO DE ITAJUBÁ