

HEMA PLENO RIO ANIL

CURSO: Desenvolvimento de Sistemas

PROF(A): Arthur Silva

DISCIPLINA: PROGRAMAÇÃO ORIENTADA A OBJETOS

ALUNO(A): _____

Tutorial de Python

Este tutorial cobre os conceitos básicos da linguagem Python, incluindo variáveis, estruturas condicionais, loops, funções e muito mais.

1. Variáveis

Em Python, variáveis são usadas para armazenar dados. Não é necessário declarar o tipo de uma variável explicitamente; o Python infere o tipo com base no valor atribuído.

Exemplo:

```
# Atribuição de valores a variáveis
nome = "João" # String/texto
idade = 25    # Inteiro
altura = 1.75 # Float
is_estudante = True # Booleano (verdadeiro ou falso)
```

```
# Exibindo valores
print(nome)
print(idade)
print(altura)
print(is_estudante)
```

Tipos de Dados Comuns:

- Inteiro (`int`): Números inteiros, como `10`, `-5`, `0`.
- Float (`float`): Números decimais, como `3.14`, `-0.001`.
- String (`str`): Texto, como `"Olá, mundo!"`.
- Booleano (`bool`): `True` ou `False`.

2. Estruturas Condicionais: `if`, `elif`, `else`

Estruturas condicionais permitem que você execute diferentes blocos de código com base em condições.

Exemplo:

```
idade = 18

if idade < 18:
    print("Menor de idade")
elif idade == 18:
    print("Tem exatamente 18 anos")
else:
    print("Maior de idade")
```

Regras:

- `if`: Verifica a primeira condição.

- ``elif``: Verifica condições adicionais se a condição do ``if`` for falsa.
- ``else``: Executa se todas as condições anteriores forem falsas.

3. Loop ``for``

O loop ``for`` é usado para iterar sobre uma sequência (como uma lista, string, etc.).

Exemplo:

```
# Iterando sobre uma lista
frutas = ["maçã", "banana", "laranja"]
```

```
for fruta in frutas:
    print(fruta)
```

```
# Iterando sobre uma string
for letra in "Python":
    print(letra)
```

Função ``range()``:

A função ``range()`` gera uma sequência de números, útil para loops.

```
for i in range(5): # 0, 1, 2, 3, 4
    print(i)
```

4. Loop ``while``

O loop ``while`` executa um bloco de código enquanto uma condição for verdadeira.

Exemplo:

```
contador = 0

while contador < 5:
    print(contador)
    contador += 1 # Incrementa o contador
```

Cuidado com loops infinitos:

Certifique-se de que a condição do ``while`` eventualmente se torne falsa, caso contrário, o loop nunca terminará.

5. Listas

Listas armazenam coleções ordenadas de itens (podem ser de tipos diferentes).

Exemplo:



```
lista = [1, "Python", 3.14,  
True]
```

```
print(lista) # Saída: [1, 'Python', 3.14, True]
```

Operações Básicas com Listas:

Operação	Exemplo	Descrição
Acesso	lista[0]	Retorna o primeiro elemento
Acesso	lista[1]	Retorna o segundo elemento
Tamanho	len(lista)	Retorna o tamanho da lista

6. Operadores Aritméticos

Operador	Descrição	Exemplo
+	Adição	5 + 3 → 8
-	Subtração	10 - 2 → 8
*	Multiplicação	4 * 3 → 12
/	Divisão	10 / 2 → 5.0
%	Módulo (resto)	10 % 3 → 1

Atividades:

1) Considere a seguinte lista em Python:
`países = ["Japão", "Brasil", "Equador"]`

Qual é o comando correto em Python para adicionar a string "Holanda" ao final desse vetor?

a) `numeros.append(Holanda)`

b) `países.add("Holanda")`

c) add in lista: `países = "Holanda"`

d) `numeros.append("Holanda")`

e) Nenhuma das alternativas

2) Dado o código Python abaixo, qual será a saída impressa quando $x = 9$?

```
x = 0
```

```
while x < 8:
```

```
    if x > 3:
```

```
        print("Maior que 3")
```

```
    elif x > 5:
```

```
        print("Maior que 5")
```

```
    elif x > 7:
```

```
        print("Maior que 7")
```

```
    x += 1
```

- a) Maior que 3
- b) Maior que 5
- c) Maior que 7
- d) Vai dar erro
- e) Nenhuma das anteriores

3) Qual condicional verifica se um número é positivo, negativo ou zero? Indique quais são as duas condicionais que faltam no código:
`num = 100`

```
if _____:  
    print("Positivo")
```

```
elif _____:  
    print("Negativo")
```

```
else:  
    print("Zero")
```

- a) `num > 0, num < 0`
- b) `num >= 0, num <= 0`
- c) `num == 0, num > 0`
- d) `num < 0, num > 0`
- e) Nenhuma das alternativas

4) Qual é a saída do seguinte código?

```
for i in range(3):  
    print(i * 2)
```

- a) 0, 1, 2
- b) 0, 2, 4
- c) 2, 4, 6
- d) 0, 2, 4, 6
- e) Nenhuma das anteriores

5) O que faz o seguinte código?

```
n=10
```

```
while n > 0:
```

```
    if n % 2 == 1:  
        print(n)  
    n = n - 1
```

- a) Imprime todos os números de 10 a 1
- b) Imprime somente os números pares de 1 a 10
- c) Imprime somente os números ímpares de 10 a 1
- d) Loop Infinito
- e) Nenhuma das alternativas

Gabarito:

- 1) d
- 2) a
- 3) a
- 4) b
- 5) c