

# Introdução a Programação Orientação a Objetos I

Código da turma no classroom: iqmtxhd

Aulas: Segunda - 20:20 e Quinta - 18:30

# Fundamentos de Python

# O QUE TEREMOS NA AULA DE HOJE!



# Tipos de Variáveis

Python é uma linguagem dinamicamente tipada, o que significa que não é necessário declarar o tipo de variável ou fazer casting (mudar o tipo de variável), pois o Interpretador se encarrega disso para nós!

Isso significa também que o tipo da variável poder variar durante a execução do programa.

**Os tipos de dados  
padrão do Python são:**

- Inteiro (`int`)
- Ponto Flutuante ou Decimal (`float`)
- Tipo Complexo (`complex`)
- String (`str`)
- Boolean (`bool`)
- List (`list`)

## **Inteiro (int)**

O tipo inteiro é um tipo composto por caracteres numéricos (algarismos) inteiros.

É um tipo usado para um número que pode ser escrito sem um componente decimal, podendo ter ou não sinal, isto é: ser positivo ou negativo.

## **Exemplos:**



```
idade = 27
```

```
ano = 2022
```

```
print(type(idade))
```

```
print(type(ano))
```

## Ponto Flutuante ou Decimal (float)

É um tipo composto por caracteres numéricos (algarismo) decimais.

Ou seja, números que podem ser representados por uma fração.

Exemplos:



```
altura = 1.80  
peso = 73.55
```

```
print(type(peso))  
print(type(altura))
```

## Complexo (complex)

Esse tipo normalmente é usado em cálculos geométricos e científicos.

Um tipo complexo contém duas partes: a parte real e a parte imaginária, sendo que a parte imaginária contém um **j** no sufixo.

## Exemplos:



```
a = 5+2j
```

```
b = 20+6j
```

```
print(type(a))
```

```
print(type(b))
```

```
print(complex(2, 5))
```



## String (str)

É um conjunto de caracteres dispostos numa determinada ordem, geralmente utilizada para representar palavras, frases ou textos.

## Exemplos:



```
nome = 'Delci'  
profissao = 'Engenheiro de Software'  
  
print(type(profissao))  
print(type(nome))
```

## Boolean (bool)

Tipo de dado lógico que pode assumir apenas dois valores: falso ou verdadeiro (False ou True em Python).

Na lógica computacional, podem ser considerados como 0 ou 1.

## Exemplos:




```
fim_de_semana = True  
feriado = False
```

```
print(type(fim_de_semana))  
print(type(feriado))
```

## Listas (list)

Listas agrupam um conjunto de elementos variados, podendo conter: inteiros, floats, strings, outras listas e outros tipos.

Exemplos:



```
alunos = ['Andreysson ', 'Delcimar ', 'Gian', 'Hercules']
notas = [10, 10, 10, 10]

print(type(alunos))
print(type(notas))
```

# Estruturas condicionais

É muito comum na programação, precisarmos que determinado bloco de código seja executado apenas se determinada condição for satisfeita.

Para esses casos, Python disponibiliza formas de se controlar o fluxo de execução de programas: são as chamadas **Estruturas Condicionais**.

# Estruturas condicionais no Python

Um conceito muito importante para se trabalhar com condições em Python é entender o conceito dos Operadores: `is`, `not`, `!=`, `==`, `in` entre outros.

## Os tipos de Estruturas Condicionais são:

- Estrutura Condicional `if`
- Estrutura Condicional `if/else`
- Estrutura Condicional `if-elif-else`
- Estrutura Condicional Ternária (`if` em uma linha)

## Estrutura Condicional if

Sua sintaxe é bem simples, bastando utilizarmos `if` seguido pela condição seguido por dois pontos:

Exemplos:



```
valor = 10
```

```
if valor > 5:  
    print('O valor é maior que 5.')
```

## Estrutura Condicional if/else

Bom, basta utilizarmos a estrutura condicional else!

Com ela, toda vez que uma condição não for atendida, o Python executará o bloco de código definido abaixo da cláusula else.

## Exemplos:



```
idade = 20


if idade < 17:
    print('A idade é MENOR que 17')
else:
    print('A idade é MAIOR que 17')
```



## Estrutura Condicional if-elif-else

O elif é utilizado quando mais de uma condição if precisa ser testada.

### Exemplos:



```
linguagem = "Python"

if linguagem == "C++":
    print('C++ é uma linguagem de programação compilada.')
elif linguagem == "Python":
    print("Python é uma linguagem de programação de alto nível")
elif linguagem == "Java":
    print("Java é uma linguagem de programação amplamente utilizada no mercado")
else:
    print('Não é nenhuma das duas opções')
```

## Estrutura Condicional Ternária (if em uma linha)

Python provê uma forma concisa de se testar valores com apenas uma linha de código

### Exemplos:



```
velocidade = 75  
  
resultado = 'Multado' if velocidade > 60 else 'Dentro do limite'  
  
print(resultado)
```

# Loops e estruturas de repetição

As estruturas de repetição são recursos das linguagens de programação responsáveis por executar um bloco de código repetidamente através de determinadas condições específicas.

## Os tipos de loops e estruturas de repetição:

- Loops utilizando for
- Loops utilizando while
- Auxiliares
- Estruturas de Repetição com Estruturas Condicionais

## Loops utilizando for

O for é utilizado para percorrer ou iterar sobre uma sequência de dados (seja esse uma lista, uma tupla, uma string), executando um conjunto de instruções em cada item.

### Exemplos:



```
computador = ['Processador', 'Teclado', 'Mouse']  
  
for item in computador:  
    print(item)
```

## Loops utilizando while

O while é uma estrutura de repetição utilizada quando queremos que determinado bloco de código seja executado **ENQUANTO** (do inglês while) determinada condição for satisfeita.

### Exemplos:



```
contador = 0
```

```
while contador < 10:  
    print(f'Valor do contador é {contador}')  
    contador += 1
```

## Auxiliadores

Existem 3 comandos que nos auxiliam quando queremos alterar o fluxo de uma estrutura de repetição.

São eles: break, continue e pass.

### Auxiliador break

É usado para finalizar um loop, isto é, é usado para parar sua execução.

### Auxiliador continue

Ao invés de encerrar o loop ele pula todo código que estiver abaixo dele (dentro do loop)

## Auxiliador `pass`

O `pass` nada mais é que uma forma de fazer um código que não realiza operação nenhuma.

## Função `enumerate()`

Ela nos entrega um contador embutido no próprio `for`!

## A função `range()`

A função `range` é de grande ajuda quando o tema é repetição, laços, `for` etc...



## Exemplos:



```
computador = ['Processador', 'Teclado', 'Mouse']  
for indice in range(len(computador)):  
    print(f"Índice={indice} | valor={computador[indice]}")
```

## Exemplos:




```
contador = 0
computador = ['Processador', 'Teclado', 'Mouse']

for elemento in computador:
    print(f"Índice={contador} | Valor={elemento}")
    contador += 1
```

# Estruturas de Repetição com Estruturas Condicionais

Podemos juntar as Estruturas de Repetição com as Estruturas Condicionais, que aliás trabalham muito bem juntas.

Exemplos:



```
for numero in range(1, 5):  
    if numero % 2 == 0:  
        print(f'0 número {numero} é par')  
    else:  
        print(f'0 número {numero} é impar')
```

# Praticando os conceitos da Aula

**Realizar o cálculo  
de salário da  
atividade anterior**

**Vamos fazer em sala.**

## **Base Cargos:**

Diretoria: R\$ 2000,00

Técnico: R\$ 1000,00

## **Processamentos:**

- Calcula a idade
- Calcula o salário  
(cargos \* tempo  
admissão (anos))

# Introdução a Programação Orientação a Objetos I

Código da turma no classroom: iqmtxhd

Aulas: Segunda - 20:20 e Quinta - 18:30

[sarahsoares.com.br/](https://sarahsoares.com.br/)