

# Fundamentos de Python 1

Jéssika Ribeiro

**mentorama.**

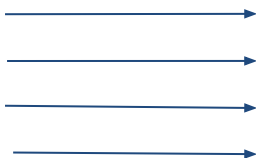
# Variáveis

# Variáveis

Uma variável é um objeto, um espaço em memória que utilizamos para representar um valor, que será usado no código.

## Variáveis

```
Nome = "Jessika"  
Idade = 28  
Altura = 1.68  
e_matematica = True
```



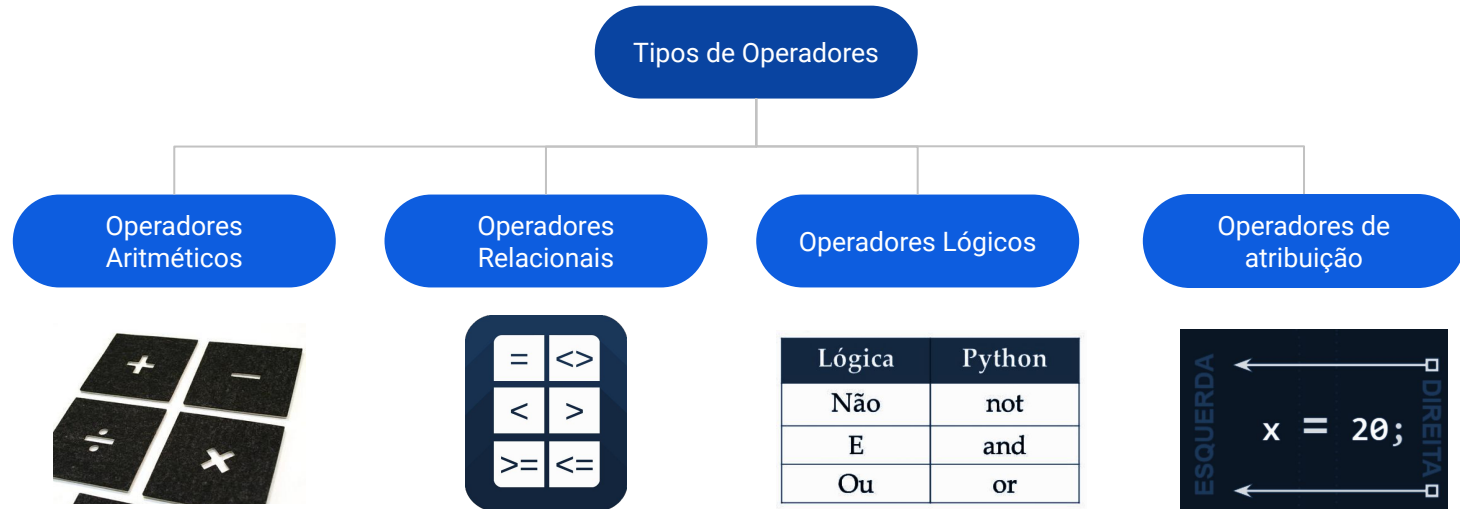
## Tipos

```
String  
Int  
Float  
Boolean
```

# Operadores

# Operadores

Formas de se combinar elementos básicos da linguagem para formar expressões mais complexas.



vamos codar?

# Estruturas de dados

# Estruturas de dados

Coleções de dados com características específicas

**Listas:** `[1,2.5, "dados",[4,8]]` → Mutáveis, dados de vários tipos;

**Tuplas:** `(1,2,3,2)` → Imutável (alterar a tupla toda apenas) , dados de vários tipos;

**Dicionários:** `{"AI":1, "ML":2, "DL":[3,4]}` → Mutáveis, conceito chave-valor;

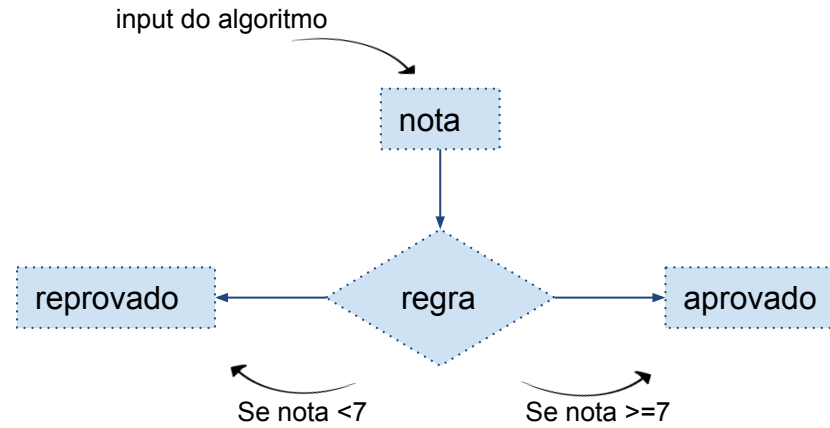
**Conjuntos:** `{1,2,3}` → Mutáveis, não ordenado, elementos únicos;

# Estruturas de decisão

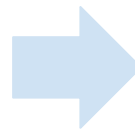
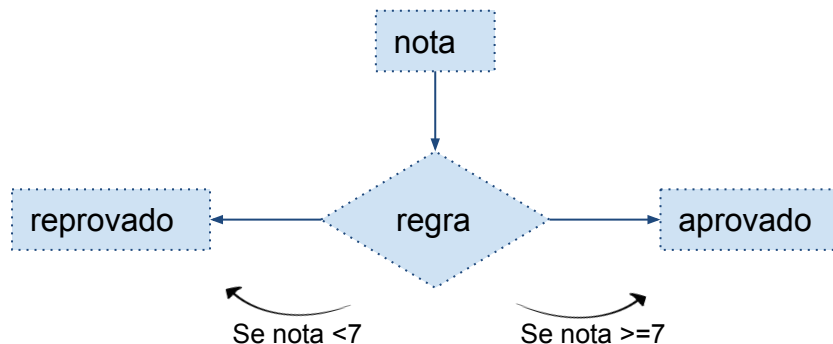


# Estruturas de decisão

Utilizadas quando queremos que o algoritmo tome caminhos diferentes de acordo com o valor de alguma variável de interesse



# If - Else



```
nota = 5

if nota >= 7:
    print("Parabéns! Você está aprovado!")
else:
    print("Você foi reprovado! :/" )
```

```
if (condição logica):
    executa se for verdadeiro
else:
    executa se for falso
```

Se a condição for verdadeira, executa o bloco indentado abaixo do if

Senão executa o bloco indentado abaixo do else

**vamos codar?**

# Estruturas de repetição

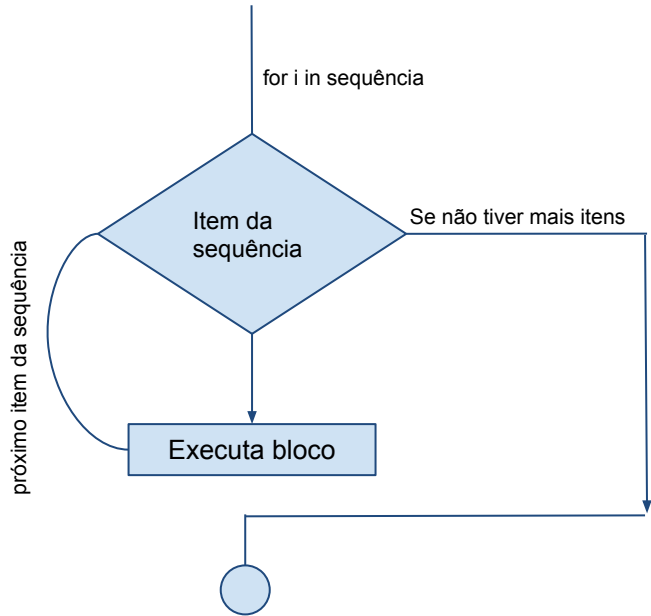
# Estruturas de repetição

As estruturas de repetição são utilizadas quando queremos que um bloco de código seja executado mais de uma vez.



# Loop For

Usado quando queremos executar um bloco de código um número fixo de vezes.



sequência

1º loop

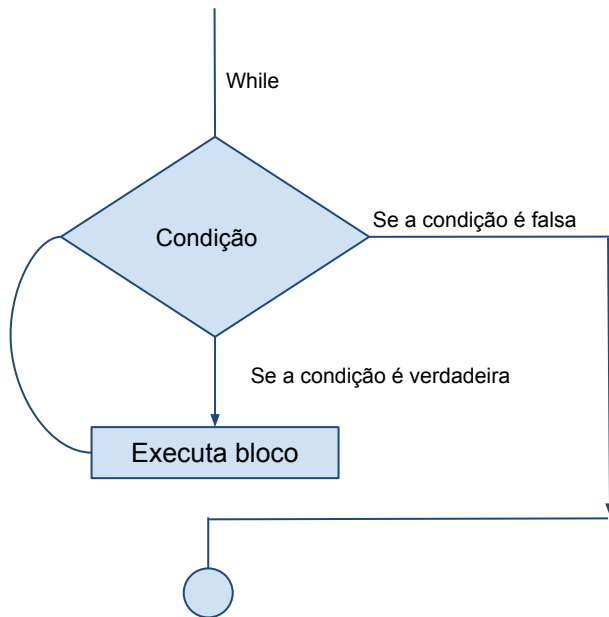
2º loop

```
notas = [9, 9.5, 8, 8.5, 7, 6.5]
var iteradora
for nota in notas:
    print(nota)
```

The code snippet shows a list `notas` with values `[9, 9.5, 8, 8.5, 7, 6.5]`. A variable `var iteradora` is declared. The `for` loop iterates over each item in `notas`, printing each item. The first iteration is labeled "1º loop" and the second iteration is labeled "2º loop".

# While

É usado quando queremos repetir um bloco de código enquanto uma condição é verdadeira



Condição do while

```
i = 1
while i < 6:
    print(i)
    i += 1
```

i = 1  
i = 2  
i = 3  
i = 4  
i = 5  
i = 6

vamos codar?

# Métodos e funções

# Métodos e funções

Métodos e funções são nossos velhos conhecidos...

```
linguagens.append(["Java", "C++"])  
linguagens
```

```
['Python', 'SQL', 'R', 'Java', 'C++']
```

```
: primeiro_dict.update({"Joaquim": 7.7})
```

```
: primeiro_dict
```

```
: {'Joao': 9, 'Maria': 7.5, 'Pedro': 6, 'Mar
```

Blocos de código reutilizáveis, ou seja, que podem ser chamados em qualquer parte do código

**Métodos:** Um bloco de instrução, com nome único e que nunca retorna valores.

```
a = print("Ola Mundo")
```

```
Ola Mundo
```

```
a
```

**VS**

**Funções :** Um bloco de instrução, com nome único e que sempre retorna valores.

```
a = len("Ola Mundo")
```

```
a
```

```
9
```



# Métodos e funções

## Built-ins

- `abs(object)`
- `bool(object)`
- `dict(k1=v1)`
- `dir(object)`
- `divmod(x, y)`
- `enumerate(iter)`
- `file(na, mod, buff)`
- `float(object)`
- `help(object)`
- `int(object)`
- `isinstance(obj, cls)`
- `len(iter)`
- `list(iter)`
- `long(object)`
- `max(iter)`
- `min(ter)`
- `open(na, mod, buff)`
- `range(start, stop, step)`
- `reversed(iter)`
- `set(iter)`
- `sorted(iter)`
- `str(object)`
- `sum(iter)`
- `tuple(iter)`
- `type(object)`
- `unicode(object)`

## Customizável

Definição da função inicia com "def"

Nome da função

Argumentos

Identação

```
def funcao_que_faz_algo(arg1, arg2):  
    """Texto de documentação"""  
    linha1  
    linha2  
    return alguma_coisa
```

"return" indica o retorno da função

vamos codar?

# Tarefa do Módulo

**Exercício 1** - Escreva o código para o seguinte padrão:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

**Exercício 2** - Escreva uma função que recebe uma lista de strings, identifica quais palavras tem tamanho maior que 7 e possuem a letra "e" e retorna uma nova lista com as palavras as quais foram identificadas com as características acima.

**Exercício 2** - Crie um dicionário com 3 itens, onde as chaves são estados brasileiros e os valores são listas de tamanho 2, contendo as duas maiores cidades desses estados.