

# Pensamento Computacional e Lógica de Programação

Prof. Dr. Álvaro Campos  
Ferreira

# Variáveis

# Operadores

# Operadores de comparação

<code>==</code>	Igual	<code>x == y</code>
<code>!=</code>	Diferente	<code>x != y</code>
<code>&gt;</code>	Maior que	<code>x &gt; y</code>
<code>&lt;</code>	Menor que	<code>x &lt; y</code>
<code>&gt;=</code>	Maior ou igual	<code>x &gt;= y</code>
<code>&lt;=</code>	Menor ou igual	<code>x &lt;= y</code>

# Operadores lógicos

**and** Verdadeiro se ambas as condições forem verdadeiras

$x < 5$  and  $x < 10$

**or** Verdadeiro se alguma das condições for verdadeira

$x < 5$  or  $x < 4$

# Operadores lógicos

**not** Inverte o resultado

`not(x < 5 and x < 10)`

# Operadores de identidade e pertencimento

**is** Verdadeiro se forem o mesmo objeto

$x \text{ is } y$

**is not** Verdadeiro se não forem o mesmo objeto

$x \text{ is not } y$

# Operadores de identidade e pertencimento

**in** Verdadeiro se o valor está presente no objeto

$x \text{ in } y$

**not in** Verdadeiro se o valor não está presente no objeto

$x \text{ not in } y$



# Listas

# Dicionários

# Dicionários

- Estrutura de dados relacional
- Acessa valores através de palavras-chave

Declarando um dicionário vazio:

```
meuDict = {}
```

# Dicionários

Declarando um dicionário com duas entradas e acessando cada uma delas:

```
meuDict = {'linguagem': 'Python',  
'idade': '28 anos'}
```

```
meuDict['linguagem']
```

```
meuDict['idade']
```

# Estruturas compostas

- Listas de dicionários
- Dicionário de listas
- Dicionário de dicionários

# Controle de fluxo

# Laços

Em português, chamamos “loop” de “laço”

– Palavra-chave **for**

```
for cliente in clientes:  
    print(cliente)
```

# Laços

Em português, chamamos “loop” de “laço”  
– Palavra-chave **for**

```
for cliente in clientes:
```

```
    print(cliente)
```

```
    if cliente == “Álvaro”:
```

```
        print(“Te peguei!”)
```





INSTITUTO BRASILEIRO DE ENSINO,  
DESENVOLVIMENTO E PESQUISA