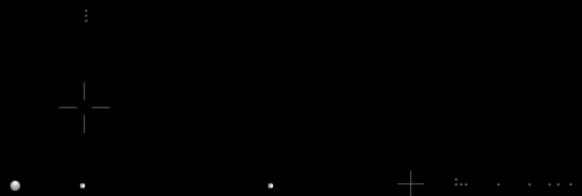
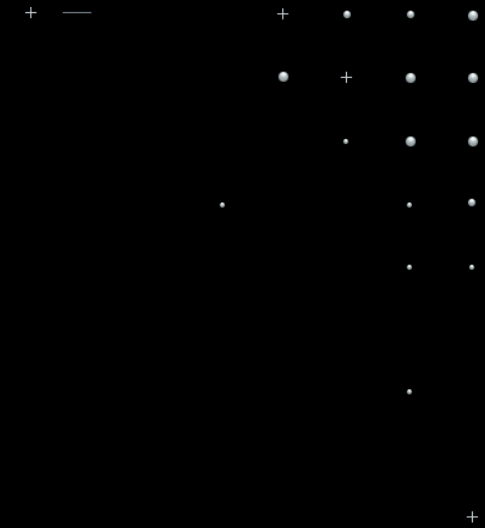


# Aula 8

## COMPUTATIONAL THINKING USING PYTHON

## AGENDA

1. Valores booleanos e expressões booleanas
2. Precedência de Operadores
3. Variáveis
4. Notações utilizadas para comentário em Python
5. Comando de Seleção
6. Exercícios



## VALORES BOOLEANOS E EXPRESSÕES BOOLEANAS

O tipo em Python para armazenar valores verdadeiro e falso é chamado bool.

Existem apenas dois valores booleanos (True e False).

***Importante: As iniciais são maiúsculas.***

## VALORES BOOLEANOS E EXPRESSÕES BOOLEANAS

Uma **expressão booleana** é uma expressão que resulta em um valor booleano.

**Importante:**

**=** *significa atribuição*

**==** *operador de comparação*

## VALORES BOOLEANOS E EXPRESSÕES BOOLEANAS

*Erro comum:*

*Como verificar se uma variável é igual a 2 ou 3 ou 5?*

*`numero == 2 or 3 or 5`*

*Está correto????*

*Não!!!!*

## VALORES BOOLEANOS E EXPRESSÕES BOOLEANAS

***Erro comum:***

***Como verificar se uma variável é igual a 2 ou 3 ou 5?***

***O operador or precisa juntar a verificação das três igualdades.***

***Correto:***

***numero == 2 or numero == 3 or numero == 5***

PRECEDÊNCIA DE OPERADORES

Nível	Categoria	Operadores
7 (alto)	Exponenciação	**
6	Multiplicação e Divisão	*, /, //, %
5	Adição e Subtração	+, -
4	Relacional	==, !=, <=, >=, >, <
3	Lógico	not
2	Lógico	and
1 (baixo)	Lógico	or

## VARIÁVEIS

### Importante

"Variáveis são um dos recursos mais básicos das linguagens de programação. Utilizadas para armazenar valores em memória, elas nos permitem gravar e ler esses dados com facilidade a partir de um nome definido por nós."

"Assim como em outras linguagens, o Python pode manipular variáveis básicas como strings (palavras ou cadeias de caracteres), inteiros e reais (float). Para criá-las basta utilizar um comando de atribuição, que define seu tipo e seu valor."



## VARIÁVEIS

Exemplo:

```
>>> mensagem = "FIAP - 1TDS"
```

```
>>> numero = 126
```

```
>>> total = 16.89
```

```
+  —  +  •  •  •  
      •  +  •  •  
      •  •  •  
      •  •  •  
      •  •  
      •  
      •  
      •  
      •  
      •
```

```
⋮  
+  
•  •  •  +  •  •  •
```

## VARIÁVEIS

Veja que não é necessário fazer uma declaração explícita de cada variável, indicando o tipo, pois isso é definido pelo valor que ela armazena.

```
>>> type(mensagem)
```

```
<class 'str'>
```

```
>>> type(numero)
```

```
<class 'int'>
```

```
>>> type(total)
```

```
<class 'float'>
```

## VARIÁVEIS

Para exibir o conteúdo dessas variáveis:

```
>>> print(mensagem)
```

FIAP - 1TDS

```
>> print(numero)
```

126

```
>>> print(total)
```

16.89

## VARIÁVEIS

### Regras para nomeação de variáveis em Python:

1. Nomes de variáveis em Python podem começar com letra ou um underline.

```
encontrou = True #válido
```

```
_encontrou = True #válido
```

```
1encontrou = True #SyntaxError: invalid syntax
```

```
$encontrou = True #SyntaxError: invalid syntax
```

## VARIÁVEIS

2. O restante do nome da variável pode ter letras, números e underline.

3. Os nomes de variáveis são sensíveis a maiúsculas.

```
numero = 10
```

```
total = Numero * 8
```

```
# NameError: name 'Numero' is not defined.
```

# VARIÁVEIS

## 4. Palavras reservadas do Python:

'False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await',  
'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except',  
'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda',  
'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with',  
'yield'

## NOTAÇÕES UTILIZADAS PARA COMENTÁRIO EM PYTHON

**Caractere cerquilha # - demarca que tudo que estiver a frente desse caractere, deve ser ignorado pelo interpretador do Python.**

**Para demarcar um conjunto de linhas é usado 3 aspas simples, ou então, 3 aspas duplas.**

## COMANDO DE SELEÇÃO

O comando de seleção é essencial para qualquer software, é ele que permite que o programa desvie seu fluxo de execução de acordo com os valores dos dados.

Se a expressão lógica for verdadeira, o fluxo segue por um caminho, senão segue por outro.

Importante: A existência do senão é opcional.



## COMANDO DE SELEÇÃO

É possível subordinar mais de uma instrução ao comando “se” ou ao comando “senão”, mas cada linguagem tem sua sintaxe particular.

Para delimitar os blocos:

Em Python é feito com deslocamento horizontal de linhas, ou seja, uma indentação.

Em Java deve-se usar abre e fecha chaves.

# COMANDO DE SELEÇÃO

Exemplo:

Linguagem	Apenas um comando	Vários comandos
Java	<pre>if(exp) comando1; else comando2;</pre>	<pre>if(exp) {comando1; comando2; comando3;} else {comando4; comando5;}</pre>
Python	<pre>if(exp): comando1 else: comando2</pre>	<pre>if(exp) :     comando1     comando2     comando3 else:     comando4     comando5</pre>

Alto-nível  
dados de

# COMANDO DE SELEÇÃO

Linguagem	if aninhado
Java	<pre>if (a&lt;b) { //comentário     if (b&lt;c) {         comando1;         comando2;     } } else if (b&gt;=c) {     comando3;     comando4; } comando5;</pre>

# COMANDO DE SELEÇÃO

Linguagem	if aninhado
Python	<pre>if (a&lt;b): #comentário     if (b&lt;c):         comando1         comando2     elif (b&gt;=c):         comando3         comando4 comando5</pre>

## EXERCÍCIOS

1) A prefeitura de Recife criou um programa de empréstimo para seus funcionários com desconto em folha. O valor da prestação não pode ultrapassar 30% do salário bruto do funcionário. Faça um programa em Python que solicite o valor do salário bruto, o valor da prestação e informe se o empréstimo pode ou não ser concedido.

Validações:

Não aceitar salário  $\leq 0$

Não aceitar prestação  $\leq 0$

# EXERCÍCIOS

2) Uma companhia aérea define os preços de suas passagens a partir da informação do destino bem como do número de passagens (se é apenas ida ou se inclui também a volta). Faça um programa em Python que solicite o destino bem como se o cliente deseja somente ida ou ida e volta. Informe o preço de acordo com a tabela abaixo (PS: a empresa não trabalha nos trechos sul e sudeste).

Região	Ida	Ida e Volta
Norte	R\$280	R\$400
Nordeste	R\$380	R\$628
Centro-Oeste	R\$620	R\$1100

## EXERCÍCIOS

3) Uma instituição bancária utiliza um dígito verificador para validar o número da conta (com 3 dígitos) de seus clientes. Faça um programa em Python que solicite o número da conta e calcule o dígito verificador. Os passos para calcular o dígito verificador são:

Ex: Número da conta = 235.

1) somar o número da conta com o seu inverso. Ex:  $235 + 532 = 767$

2) Multiplicar cada dígito do número obtido no passo anterior pela sua ordem posicional e somar esses resultados. O último dígito do número obtido é o dígito verificador.

– Ex:  $7 \times 1 + 6 \times 2 + 7 \times 3 = 40$  (dígito verificador  $\rightarrow 0$ ).

Observação: Use o operador % e a divisão inteira para obter o dígito verificador.



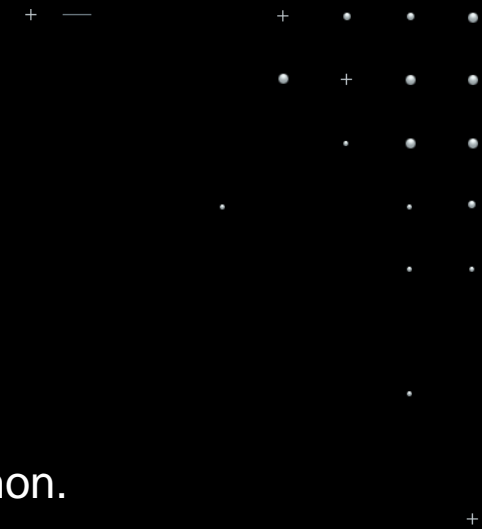
## EXERCÍCIOS

**4) Faça o teste de mesa para numero\_1= 15 e numero\_2= 6**

```
resultado = ((numero_1%2)*3)+(13-2+numero_2)
```

Responda:

- Qual o conteúdo da variável resultado quando termina o algoritmo?
- Análise o pseudocódigo e responda: o que mostrará na tela?
- Reescrever o algoritmo utilizando a linguagem de programação Python.



## EXERCÍCIOS

```
//Pseudocódigo
var
numero_1, numero_2, resultado: inteiro
inicio
// Seção de Comandos
escreval("Digite um número inteiro: ")
leia(numero_1)
escreval("Digite um número inteiro: ")
leia(numero_2)

resultado<- ((numero_1 mod 2) * 3) + (13 - 2 + numero_2)

se(resultado<=0) entao
    escreval("Resultado menor ou igual a zero")
senao
    se((resultado>0) e (resultado<=20)) entao
        escreval("Resultado maior que zero e menor ou igual a 20")
    senao
        escreval("Resultado maior que 20")
    fimse
fimse
finalgoritmo
```

