

## Estruturas de controle

Estruturas de controle permitem que possam ser realizadas interações/iterações com os dados fornecidos ou produzidos por um programa de computador. Portanto, na etapa de processamento, é possível utilizá-las para produzir cada vez programas e algoritmos mais complexos. Estas estruturas estão divididas em:

- Desvios condicionais
- Laços de repetição

### Desvios condicionais

Como o próprio nome diz, estruturas de desvios condicionais permitem criar condições para a realização de um determinado procedimento em um programa. Estas decisões são comuns no cotidiano, por exemplo:

- Um indivíduo com 18 anos ou mais é considerado maior de idade no Brasil;
- O evento irá ocorrer somente se não chover; e
- Caso o usuário fale espanhol, a interface do seu sistema deve apresentar instruções nesta língua.

Em todas estas situações, e em tantas outras, a condição é representada, ou ao menos pode ser formalizada, através de uma condição **SE**:

- Se o indivíduo tem 18 anos ou mais será considerado maior de idade no Brasil;
- Se não chover, o evento irá ocorrer; e
- Se o usuário compreende apenas a língua espanhola interface deve apresentar instruções em espanhol.

Quando convertemos o exemplo da maioria para um programa de computador, obtemos a seguinte estrutura (Quadro 1):

Quadro 1. Estrutura de desvio condicional.

```
idade = eval(input("Informe a sua idade:"))  
if idade >= 18:  
    print("É maior de idade.")  
else:  
    print("É menor de idade.")
```

No exemplo do Quadro 1 identificam-se três novas palavras ou símbolos reservados da linguagem Python:

- **IF**: traduzido para português significa SE;
- **ELSE**: traduzido para português significa SENÃO; e
- **:**: indica o que deve acontecer se aquela condição for satisfeita.

É importante ressaltar que é a indentação que irá indicar quais comandos estão dentro de qual. No exemplo do Quadro 1, o programa somente irá imprimir "É maior de idade" se a idade informada for maior ou igual à 18. Seguindo, se esta idade não atender aquela condição, então será impresso "É menor de idade".

## Desvios condicionais aninhados

É possível que um programa/problema esteja relacionado a mais de uma condição. É possível implementar esta situação através de desvios condicionais aninhados (Quadro 2). Como exemplo, podemos pensar em um programa que verifica se um número está dentro de um intervalo:

- Imprima A, se o número for  $< 3$  (menor que 3);
- Imprima B, se o número for 3; e
- Imprima C, se o número for  $> 3$  (maior que 3).

Quadro 2. Estrutura de desvio condicional aninhado.

```
numero = eval(input("Informe um número:"))  
if numero < 3:  
    print("A")  
elif numero == 3:  
    print("B")  
else:  
    print("C")
```

Para que uma condição seja satisfeita, deve-se esperar um retorno Booleano do teste (True ou False). As comparações possíveis são apresentadas na Tabela 1:

Significado	Operador Python	Exemplo	Resultado
Menor que	$<$	$1 < 2$	True
Maior que	$>$	$1 > 2$	False
Menor ou igual a	$\leq$	$1 \leq 2$	True
Maior ou igual a	$\geq$	$1 \geq 2$	False
Igual a	$==$	$1 == 2$	False
Diferente de	$!=$	$1 != 2$	True

Além disso, utilizando a lógica Booleana, existem dois operadores que permitem fazer condições mais complexas, o AND (e) e o OR (ou). No Quadro 3 é possível ser visto um exemplo de seu uso.

Quadro 2. Estrutura de desvio condicional aninhado.

```
numero = eval(input("Informe um número:"))  
if numero < 3:  
    print("A")  
elif numero == 3:  
    print("B")  
else:  
    print("C")
```