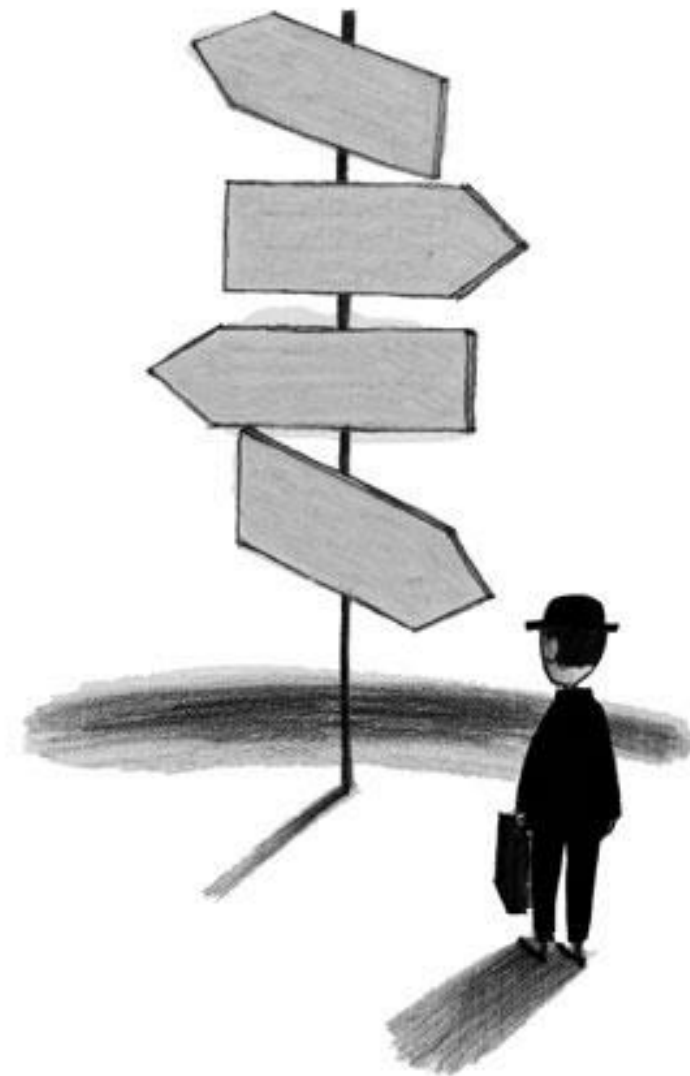


Estrutura de Decisão

≡ Estrutura de Decisão ou Condicional

1. Ponto de tomada de **decisão**
2. Permite a escolha do grupo de ações e estruturas a ser executado quando **determinadas condições** (expressões lógicas) **são ou não satisfeitas**
3. A escolha faz com que você caminhe por um **fluxo** ou por outro



≡ Exemplos de Situações Condicionais

- Verificar se um aluno passou por média;
- Verificar se um aluno está aprovado em uma disciplina;
- Verificar se um número é primo;
- Verificar se uma pessoa pode tirar a CNH no Detran;
- ...



Expressão Lógica

- O resultado de uma expressão lógica será sempre um valor tipo booleano (“bool”):
 - Verdadeiro (“True”)
 - Falso (“False”)
- Pode ser formada por:
 - Variáveis ou constantes tipo “bool”
 - Operadores Relacionais
 - Operadores Lógicos

≡ Operadores Relacionais

Operador	Descrição	Exemplo	Resultado
>	Maior que	$2 > 2$	False
<	Menor que	$2 < 2$	False
>=	Maior igual que	$2 >= 2$	True
<=	Menor igual que	$2 <= 2$	True
!=	Diferente	$2 != 2$	False
==	Igual	$2 == 2$	True

≡ Operadores Lógicos

- A expressão lógica pode conter também **operadores lógicos**:
and / **or** / **not**

and		
Cond.1	Cond.2	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

or		
Cond.1	Cond.2	Resultado
V	V	V
V	F	V
F	V	V
F	F	F

not	
Cond.	Resultado
V	F
F	V

≡ Operadores Lógicos

Operador	Significado	Exemplo	Resultado
and	E	2 == 2 and 4 == 4	True
or	Ou	2 == 2 or 4 == 6	True
not	Não	not 2 == 2	False

- A prioridade dos operadores relacionais é maior do que a dos operadores lógicos.
- Dentre os operadores lógicos a prioridade é: *not*, *and*, *or*
- Para quebrar a prioridade devemos usar parênteses



Exemplos

Condição	Resultado
$2 > 3$	False
'a' > 'A'	True
'joana' > 'maria' and $10 < 20$	False

- A função **ord()** devolve o código numérico do caractere passado como parâmetro. Ex: **ord('a')** devolve **97**
- A função **chr()** devolve o caracter corresponde ao código numérico passado como parâmetro. Ex: **chr(97)** devolve **'a'**



Decisão Simples

Sintaxe:

```
if <condição>:  
    <comandos>
```

Exemplo

```
if (nota >= 7):  
    print("aprovado")
```

Observação: Os Comandos (bloco de instruções) que serão executados, se a condição for atendida, devem ser endentados corretamente, respeitando as regras da sintaxe da linguagem Python

≡ Decisão Composta

Sintaxe:

```
if <condição>:  
    <comandos>  
else:  
    <comandos>
```

Exemplo

```
if (nota >= 7):  
    print("aprovado")  
else:  
    print("reprovado")
```



Decisão Aninhada

Sintaxe:

```
if <condição>:  
    <comandos>  
else:  
    if <condição>:  
        <comandos>  
    else:  
        <comandos>
```

Exemplo

```
if (nota >= 7):  
    print("aprovado")  
else:  
    if (nota >= 4):  
        print("final")  
    else:  
        print("reprovado")
```

≡ Decisão Aninhada (usando *elif*)

Sintaxe:

```
if <condição>:  
    <comandos>  
elif <condição>:  
    <comandos>  
elif <condição>:  
    <comandos>  
...  
else:  
    <comandos>
```

Exemplo

```
if (nota >= 7):  
    print("aprovado")  
elif (nota >= 4):  
    print("final")  
else:  
    print("reprovado")
```

≡ Decisão Múltipla (match...case)

Sintaxe:

```
match <variável>:  
    case <valor>:  
        <comandos>  
    case <valor>:  
        <comandos>  
    ...  
    case _:  
        <comandos>
```

Observação: funciona apenas na versão do Python 3.10 ou posterior.

Exemplo

```
x = int(input('1º operando: '))  
y = int(input('2º operando: '))  
op = input('Operador: ')  
match op:  
    case '+':  
        print(x + y)  
    case '-':  
        print(x - y)  
    case 'x' | '*':  
        print(x * y)  
    case '/':  
        print(x / y)  
    case '%':  
        print(x % y)  
    case _:  
        print('Operador desconhecido')
```