

# Estruturas de Decisão

Introdução a Programação

# Objetivo de Aprendizagem

- Conhecer as estruturas de decisão em Python

# Agenda

- *Casting*
- `if-else`
- `if-else-elif`

# Casting

## O que é *casting*?

**Casting** é um método para converter ou forçar que uma variável seja de um determinado tipo ( `string` , `int` , `float` , etc). O Python realiza o *casting* automaticamente em algumas situações. Porém, em outros casos o programador precisa fazê-lo manualmente.

# Como fazer *casting*?

- Existem algumas funções úteis relacionadas com o *casting*
  - `type()`
  - `int()`
  - `float()`
  - `str()`

# Exemplo 1

```
1  # Python converte automaticamente a para int
2  a = 7
3  print(type(a))
4  # Python converte automaticamente b para float
5  b = 3.0
6  print(type(b))
7  # Python converte automaticamente c para float por conta da soma
8  c = a + b
9  print(c)
10 print(type(c))
11 # Python converte automaticamente d to float por conta da multiplicação
12 d = a * b
13 print(d)
14 print(type(d))
```

## Exemplo 2

```
1  # O que acontece ao executar o
2  # código abaixo?
3  idade = input('Qual sua idade?')
4  idade++
5  print('No próximo ano minha idade será', idade)
```

- O que acontece ao executar o código?
- Qual erro apresentado?
- Como corrigir?



# Exemplo 2

Corrigido

```
1  # O que acontece ao executar o
2  # código abaixo?
3
4  # a funcao int() força que idade
5  # seja do tipo int
6  idade = int(input('Qual sua idade?'))
7  # variável idade é incrementada
8  idade++
9  # a mensagem é exibida
10 print('No próximo ano minha idade será', idade)
```

- Na linha 6 é realizado o *casting*
- Agora pode-se realizar a operação de incremento (linha 8)
- Não há mais erro

# Estruturas de decisão

# Declaração Condicional

Declarações condicionais em Python são utilizadas para executar certas partes do código apenas se determinadas condições forem atendidas. Estas declarações controlam o fluxo do código.

# if

A declaração `if` executa um bloco de código caso a condição declarada seja atendida ( `True` ).

## Exemplo 3

```
1  idade = 20
2
3  if idade ≥ 16:
4      print("Cidadão pode votar.")
```

## if-else

A declaração `if-else` executa um bloco de código caso a condição declarada seja atendida ( `True` ), caso contrário outro bloco ( `else` ) é executado. **NUNCA** ambos os blocos serão executados. **SEMPRE** haverá pelo menos um dos blocos será executado.

## Exemplo 4

```
1  age = 10
2
3  if age ≤ 12:
4      print("Viajante gratuito.")
5  else:
6      print("Viajante pagante.")
```

## Exemplo 5

```
1  nota = 45
2  res = "Aprovado" if nota ≥ 60 else "Reprovado"
3
4  print(f"Result: {res}")
```



## elif

A declaração `elif` ( `else + if` ) permite a checagem de condições múltiplas e executar diferentes blocos de código de acordo com a condição verdadeira. **Cuidado**, apenas uma condição pode ser verdadeira, um único bloco de código é executado.

# Exemplo 6

```
1  idade = 25
2
3  if idade ≤ 12:
4      print("Criança.")
5  elif age ≤ 19:
6      print("Adolescente.")
7  elif age ≤ 35:
8      print("Jovem adulto.")
9  else:
10     print("Adult.")
```

## if-else aninhados

Pode-se criar estruturas de if-else dentro de outras if-else criando um aninhamento. Utiliza-se if-else aninhados para checar condições em sequência.

# Exemplo 7

```
1  idade = 70
2  e_membro = True
3
4  if idade ≥ 60:
5      if e_membro:
6          print("Desconto senior, 30%")
7      else:
8          print("Desconto senior 20%")
9  else:
10     print("Não elegível para desconto senior.")
```

# Exercícios

# 1

Modificar a condição do `if-else` para que o código dentro do `else` seja executado. A saída do *script* deve ser '2'.

## 1 (cont.)

```
1 option = 2
2
3 if option > 0:
4     var = 1
5 else:
6     var = 2
7 print(var)
```

## 2

Criar um *script* em Python que recebe dois números do usuário e a operação aritmética desejada (soma, subtração, multiplicação e divisão). Em seguida a operação é realizada com os números e o resultado exibido ao usuário.



# 3

Crie um programa que simule um sistema de login simples. O programa deve pedir ao usuário um nome de usuário e uma senha e checar se:

- O nome de usuário tem pelo menos 3 caracteres.
- A senha tem exatamente 6 dígitos (e é numérica). Se tudo estiver correto, exibir: "Login criado com sucesso!", Caso contrário, informar qual regra não foi atendida (ex: "Senha deve ter 6 dígitos numéricos!").


# Referências

- <https://www.geeksforgeeks.org/type-casting-in-python/>
- <https://www.geeksforgeeks.org/conditional-statements-in-python/>

Prof. José Roberto Bezerra

[jbroberto@ifce.edu.br](mailto:jbroberto@ifce.edu.br)

---

Powered by  Slidify