Aula: 02- Python
Assunto: if, elif, else – for – while – operadores – list comprehensions

Introdução

Python X Java

Suponha um desvio condicional simples: se x maior ou igual a zero, some dois a y e subtraia três de z.

Em Java este código fica:

```
if(x >= 0){
  y += 2;
  z -= 3;
}
```

Em Python:

```
if x >= 0:
   y += 0
   z -= 3
```

Python não tem ponto e vírgula separando comandos nem chaves para separar blocos. A separação de comandos é feita pulando linha. A identificação dos blocos é feita pela endentação.

If – Elif - Else

Forma geral do desvio condicional em Python:

```
if case1:
    perform action1
elif case2:
    perform action2
else:
    perform action3
```

Esta estrutura é parecida com a estrutura if – else if – else do Java e se comporta da mesma maneira. São válidas as combinações somente o if, if else, if else if, if else if else. Os else ifs podem se repetir indefinitamente.

Exemplos

```
if True:
    print('É verdadeiro')

x = 1
if x > 0:
    print('x é positivo')
else:
    print('x é negativo ou nulo')

x = -1
if x > 0:
    print('x é positivo')
elif x == 0:
    print('x é nulo')
else:
    print('x é negativo')

#indentação
if True:
    print('É verdadeiro')
    print('Dentro do if')
print('fora do if')

x = 0
if x > 0:
    print('x é positivo {}'.format(x))
else:
    if x == 0:
        print(f'x = {x} é nulo')
    else:
        pass
    print('x é negativo')
```

for Loops

O loop for do Python é bem parecido com o for-each do Java. Ele itera por itens iteráceis, como strings, listas, tuplas e dicionários. E sempre percorre o item do início ao fim, a não ser que a gente saia usando um **break**.

A forma geral é:

```
for item in object:
```

```
    comandos para fazer coisas

lista1 = [1,2,3,4,5,6,7,8,9,10]

#iterando em uma lista
for num in lista1:
    print(num)

#o nome da variável não faz diferença
for garrafa in lista1:
    print(garrafa)

#somente os pares
for num in lista1:
    if num%2 == 0:
        print(num)

#somar os numero da lista
list_sum = 0
for num in lista1:
    list_sum += num
print(f'A soma é {list_sum}')

#iterando em uma string
for letra in 'Esta é uma string.':
    print(letra)

#iterando em uma tupla
tup = (1,2,3,4,5)
for t in tup:
    print(t)

lista2 = [(2,4),(6,8),(10,12)]

for tup in lista2:
    print(tup)

#tuple unpacking - desempacotando
for t1,t2 in lista2:
    print(t1, t2)

d = {'k1':1, 'k2':2, 'k3':3}

#iterando em um dicionário
```

```
#neste formato traz somente as chaves
for item in d:
    print(item)

#usando unpacking dá para retornar a chave e o valor
for k,v in d.items():
    print(k)
    print(v)
```

while Loops

O loop while é semelhante ao while do Java, exceto pelo else, que é executado no final, quanto a expressão condicional do while se torna falsa e o programa sai do loop.

A forma geral do while é:

```
while test:
    code statements
else:
    final code statements
```

break, continue, pass

break: sai do loop, inclusive pulando os comandos que estão depois dele, mudando o fluxo de execução para a primeira linha do programa depois do loop.
continue: volta para o começo do loop, deixando de executar todos os comandos do loop que estão depois do continue naquela iteração.
pass: não faz nada, mas o Python não aceita statements sem comando; por exemplo, um if sem nada para fazer; neste caso, use o pass.

```
x = 0

while x < 10:
    print(f'o valor de x é {x}')
    print('x ainda é menor que 10')
    x+=1

x = 10

while x < 10:
    print(f'o valor de x é {x}')
    print('x ainda é menor que 10')
    x+=1
else:
```

```
    print("Terminou!")

x = -1
while True:
    x+=1
    if x > 10:
        print(f'{x} Saindo...')
        break
    else:
        continue
        print(f'{x} Continuando...')
```

pass: não faz nada, mas o Python não aceita statements sem comando; por exemplo, um if sem nada para fazer; neste caso, use o pass.

Operadores Úteis

range: gera uma lista de inteiros; porém, para ser uma lista mesmo, é necessário usar a função list(). Ex: list(range(1,5)).

enumerate: útil para criar uma variável contadora em um loop.

zip: cria uma lista de tuplas a partir de duas listas.

in: retorna true se um elemento estiver em um objeto.

min e **max:** verifica o maior e o menor número de uma lista

random: gera números aleatórios

input: entrada de dados via teclado

```
range(0,11)

list(range(0,11))

list(range(11))

list(range(5,11))

list(range(0,11,2))

list(range(0,101,10))
```

Exercícios resolvidos

Desvio Condicional

1. Faça um Programa que peça dois números e imprima o maior deles.

2. Faça um programa para a leitura de duas notas parciais de um aluno. O programa deve calcular a média alcançada por aluno e apresentar:
- A mensagem "Aprovado", se a média alcançada for maior ou igual a sete;
- A mensagem "Reprovado", se a média for menor do que sete;
- A mensagem "Aprovado com Distinção", se a média for igual a dez.

3. As Organizações Tabajara resolveram dar um aumento de salário aos seus colaboradores e lhe contrataram para desenvolver o programa que calculará os reajustes.
Faça um programa que recebe o salário de um colaborador e o reajuste segundo o seguinte critério, baseado no salário atual:
- salários até R$ 280,00 (incluindo) : aumento de 20%
- salários entre R$ 280,00 e R$ 700,00 : aumento de 15%
- salários entre R$ 700,00 e R$ 1500,00 : aumento de 10%
- salários de R$ 1500,00 em diante : aumento de 5% Após o aumento ser realizado, informe na tela:
- o salário antes do reajuste;
- o percentual de aumento aplicado;
- o valor do aumento;
- o novo salário, após o aumento.

4. Faça um programa que lê as duas notas parciais obtidas por um aluno numa disciplina ao longo de um semestre, e calcule a sua média. A atribuição de conceitos obedece à tabela abaixo:

Média de Aproveitamento	Conceito
Entre 9.0 e 10.0	A
Entre 7.5 e 9.0	B
Entre 6.0 e 7.5	C
Entre 4.0 e 6.0	D
Entre 4.0 e zero	E

O algoritmo deve mostrar na tela as notas, a média, o conceito correspondente e a mensagem “APROVADO” se o conceito for A, B ou C ou “REPROVADO” se o conceito for D ou E.

5. Faça um programa que calcule as raízes de uma equação do segundo grau, na forma ax2 + bx + c. O programa deverá pedir os valores de a, b e c e fazer as consistências, informando ao usuário nas seguintes situações:
- Se o usuário informar o valor de A igual a zero, a equação não é do segundo grau e o programa não deve fazer pedir os demais valores, sendo encerrado;
- Se o delta calculado for negativo, a equação não possui raizes reais. Informe ao usuário e encerre o programa;
- Se o delta calculado for igual a zero a equação possui apenas uma raiz real; informe-a ao usuário;
- Se o delta for positivo, a equação possui duas raiz reais; informe-as ao usuário;

Fonte:

Lista de Exercícios de Estrutura de Decisão;
https://wiki.python.org.br/EstruturaDeDecisao