

Fundamentos de Programação de Computadores

Python- condições, ciclos,
strings

Docente: Fátima Leal

DCT DEPARTAMENTO CIÊNCIA
E TECNOLOGIA

Python

- Na última aulas vimos:
- **Variáveis**
- **Expressões**
- **Funções**

Python – Operadores de comparação

- Como experimentamos no pseudocódigo, nas condições necessitamos de operadores de comparação.

<code>==</code>	igual
<code>!=</code>	diferente
<code>></code>	maior
<code><</code>	menor
<code>>=</code>	maior ou igual
<code><=</code>	menor ou igual

```
>>> 4+2 == 6
True
>>>
>>> 3!=4
True
>>>
```

- No Python, estes operadores são utilizados de forma semelhante.

Python – Operadores de comparação

- Adicionalmente, por vezes necessitamos de agregar condições utilizando operadores lógicos.

operador	o que se verifica
and	se ambos os operandos são verdadeiros
or	se pelo menos um dos operandos é verdadeiro
not	se o operando é falso

```
>>> 4+2==6 and 3!=4
True
```

```
>>> 4+3==6 or 3!=4
True
```

Tabela da Verdade

P	Q	P and Q
F	F	F
F	V	F
V	F	F
V	V	V

P	Q	P or Q
F	F	F
F	V	V
V	F	V
V	V	V

P	not P
F	V
V	F

Python – Comparação Strings

- Os operadores de comparação também podem ser utilizados com strings

```
>>> word = "banana"
>>> word == "apple"
False
>>> word == "banana"
True
>>> word == "Banana"
False
```

Python – condições (if and else)

A condição é seguida de :

```
if <BOOLEAN EXPRESSION>:  
    <STATEMENTS_1> # Executed if condition evaluates to True  
else:  
    <STATEMENTS_2> # Executed if condition evaluates to False
```

```
if x < y:  
    <STATEMENTS_A>  
elif x > y:  
    <STATEMENTS_B>  
else:      # x == y  
    <STATEMENTS_C>
```

=

```
if x < y:  
    <STATEMENTS_A>  
else:  
    if x > y:  
        <STATEMENTS_B>  
    else:  
        <STATEMENTS_C>
```

Abreviação do else if

Python – condições (if and else)

```
if x > 0:
    if x < 10:
        print("x é um algarismo entre 0 e 9!")

if x > 0 and x < 10:
    print("x é um algarismo entre 0 e 9!")

if 0 < x < 10:                #Possível no Python mas não em muitas linguagens
    print("x é um algarismo entre 0 e 9!")
```

Nota que temos diferentes formas de construir condições. Por vezes, a leitura do código fica mais simples sem os ses encadeados.

Python – condições (if and else)

- É possível negar condições utilizando a palavra reservado do Python **not**

```
if not (age >= 18):  
    print ("Ainda nao entrou na idade adulta!")
```

```
if age < 18:  
    print ("Ainda nao entrou na idade adulta!")
```

not (A == B)	\iff	A != B
not (A < B)	\iff	A >= B
not (A <= B)	\iff	A > B

Exercício

Constrói um algoritmo que permita calcular o preço de saldo de um artigo. Os descontos variam em função do preço, conforme se mostra na tabela abaixo.

O preço do artigo deve ser fornecido pelo utilizador.

Preço (€)	Desconto
$P > 200$	60%
$100 < P \leq 200$	40%
$50 < P \leq 100$	30%
$P \leq 50$	20%

Python – ciclos

- Computadores são frequentemente utilizados para automatizar tarefas repetitivas.
- Repetição de uma sequência de instruções: iteração.
- Iteração em Python: ciclos **for**, ciclos **while**
- O ciclo **for** permite percorrer uma lista e é utilizado quando se conhece o número máximo de vezes que se irá iterar, ou seja, **iteração definida**.
- O ciclo **while** repete uma sequência de instruções de acordo com uma condição imposta à cabeça. É utilizado quando se vai iterar até que determinada condição se verifique, ou seja, **iteração indefinida**.

Python – ciclo for

- Vamos começar a trabalhar com o ciclo for utilizando a função **range** do Python

```
for x in range (10):  
    print(x)
```

- Neste exemplo, a variável x vai tomar valores de 0 a 9. Um ciclo for **começa sempre em 0**.

Python – ciclo for

```
for x in range(5): # 0, 1, 2, 3, 4
    print(x)

for x in range(10): # 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
    print(x)

for x in range(3,10): # 3, 4, 5, 6, 7, 8, 9
    print(x)

for x in range(3,10,2): # 3, 5, 7, 9
    print(x)
```

- **range(n)** → valores inteiros de 0 até n-1 inclusive
- **range(i,n)** → valores inteiros de i até n-1 inclusive
- **range(i,n,d)** → valores inteiros i, i + d, i + 2d, . . . , inferiores a n

Python – ciclo while

- Permite iterar sem saber a priori quantas vezes se vai executar o ciclo. Utiliza-se uma condição.

```
while <CONDITION>:  
    <STATEMENTS>
```

Exemplo:

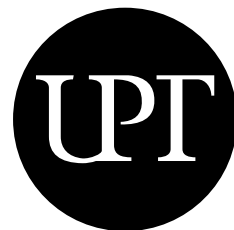
```
n=6  
current_sum = 0  
i=0  
while i <= n:  
    current_sum += i  
    i += 1  
print(current_sum)
```

Exercícios

1- Desenvolve um algoritmo que leia 10 números do tipo inteiro e determine a percentagem de números positivos e a média dos números negativos lidos.

2- A Universidade Portucalense precisa de um algoritmo que permita ler as notas de Fundamentos de Programação obtidas pelos elementos de uma turma para calcular a média.

O algoritmo termina quando for introduzido uma nota negativa. No final deve ser mostrada a média da turma.



UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.