

# Breve introdução ao Python.

(Definir variáveis, entrada e saída de dados, operadores, ciclos com condições)

## Notas:

- 1) Vamos aprender gradualmente os procedimentos da programação em Python.
- 2) Primeiro definimos a **plataforma** onde vamos programar.
- 3) Se não compreender as sugestões indicada abaixo, pode usar o tutorial que está em: <https://pedronoia.pt/python/py.htm>

## Sugestões 1:

-Se a sua calculadora gráfica tem o Python, pode tentar habituar-se a trabalhar com ela.

-Se a sua escola usa computadores com sistema Windows, pode ser prático instalar o IDLE do site <https://www.python.org>

-Se usa sistema android como tablet ou telemóvel, pode ser mais prático usar o “Colab” da Google, pois funciona bem em todos os dispositivos. <https://colab.research.google.com> Para utilizar, é necessário ter uma conta do *Gmail*.

## Sugestões 2:

Para **atribuir valores a variáveis** é necessário saber com que tipo de variável estamos a trabalhar. Por exemplo, se for um número inteiro, devemos indicar **int**, se for real é **float** as listas são **list**, as sequências de caracteres é **str** e os booleanos (verdadeiro ou falso é **bool**).

## Entrada e saída de dados

Para pedir ao utilizador para introduzir um número, usamos o **input**. Devemos atribuir um nome, que pode ser apenas uma letra, para essa entrada.

### Exemplo

```
x=input()
```

A frase que aparecerá para o utilizador fica entre aspas:

### Exemplo

```
x=input("introduza um número")
```

Para exigir que o número introduzido seja de um determinado tipo, por exemplo inteiro(*int*), colocamos essa indicação antes do input.

**Exemplo**

```
x=int(input("introduza um número"))
```

Devemos ter cuidado com os parênteses que abrem e que fecham, assim como as aspas.

Para mostrar uma saída de dados, usamos o *print*.

Para mostrar apenas uma frase, usamos essa frase entre aspas.

**Exemplo**

```
print("introduza um número inteiro")
```

Se queremos apresentar uma frase e uma variável introduzida, podemos fazer:

```
x=int(input("introduza um número"))  
print("o seu número é", x).
```

## Operadores.

**Exemplo:** Se indicarmos

```
print(3<1)
```

isto é, 3 menor que 1. O programa responde "*false*", pois, esta desigualdade é falsa. Se fosse

```
print(3>1)
```

a resposta seria "*true*".

Para somarmos usamos o símbolo + e para subtrair é -.

Na multiplicação, usamos \* e na potenciação \*\*.

**Exemplo:** Para  $2^5$  fazemos, escrevemos:

```
2**5
```

Também podemos usar *int* para a parte inteira de um número,

**Exemplo**

```
int(2.37)
```

dá 2

## **Ciclos:-if – elif - else.**

Usamos o **if** para colocar uma condição e **else** para a situação contrária.

### **Exemplo:**

Introduzo um número inteiro. Se esse número for maior que 10, aparecerá a frase “número maior que 10”, caso contrário parecerá a frase: “número menor ou igual a 10”

```
z=int(input("introduza um número inteiro")).  
If z>10:  
    Print("Número maior que 10")  
Else:  
    Print("número menor ou igual a 10")
```

Muito cuidado: depois da condição do **if**, é obrigatório os dois pontos( : ). Do mesmo modo depois do **else**. Além disso, a ação a executar deverá estar escrita na linha seguinte com um pequeno avanço para a frente. Repare com atenção para os alinhamentos das linhas de programa.

Podemos usar o **if** para colocar uma condição, o **elif** para outra condição e **else** para a última situação contrária.

### **Exemplo**

Por exemplo, introduzo um número inteiro. Se esse número for maior que 10, aparecerá a frase “número maior que 10”, se for igual a 10, aparece “número igual a 10”, se for menor que 10, aparecerá a frase: “número menor que 10”

```
z=int(input("introduza um número inteiro")).  
If z>10:  
    Print("Número maior que 10")  
elif z==10:  
    Print("Número maior que 10")  
Else:  
    Print("número menor ou igual a 10")
```

Repare que, para testar se o número é igual a 10, fazemos == e não com =.

De facto, = é para atribuir valores e == é para testar se um certo valor é igual a outro.

## Ciclos de repetição “for e o range” e o while

Vamos começar por usar o *for* e o *range*.

### **Exemplo**

Vamos escrever os números entre 0 e 3:

```
For a in range(4):  
    print(a)
```

Obteremos os números: 0; 1; 2; 3. Curiosamente começa em zero e são ao todo 4 número. Não esquecer os dois pontos (:) da primeira linha e o avanço de espaço na segunda linha.

### **Exemplo:**

Vamos somar os números 0, 1, 2, 3, 4, 5,6,7,8, 9.

Seja **s** a soma e **num** o número.

A soma, **s**, começa por ser zero, depois é adicionada ao 1, o seu resultado é adicionado ao 2, o seu resultado é adicionado ao 3, e assim sucessivamente até ao 9.

O **num** começa em zero e vai até 9.

```
s=0  
for num in range(10):  
    s=s+num  
    print( "a soma é:", s)
```

O resultado é 45, que corresponde a  $0+1+2+3+4+5+6+7+8+9=45$

Utilizemos agora o *while*.

### **Exemplo**

Vamos pegar no número 20 e retirar uma unidade, passando para 19, e assim sucessivamente até o número chegar a 10. O objetivo é obter os números:

20; 19; 18; 17; 16; 15; 14; 13; 12; 11; 10.

```
c=20  
while c>9:  
    print(c)  
    c=c-1
```

Repare que, enquanto o número é ,maior que 9, é apresentado o seu valor e depois, retirada uma unidade. Isto vai parar quando o **c** deixar de ser superior a 9.

**Nota:** para praticar, deve resolver os exemplos:

<https://pedronoia.pt/python/pypm24.htm>

**Sugestões Python:**

<https://pedronoia.pt/python/py.htm>

<https://pedronoia.pt/python/10m16.htm>

<https://pedronoia.pt/python/10m17.htm>