

Exercícios do Curso de Python

Ficha 1

Variáveis, Fluxo de controlo e Loops

1. Escreva em linguagem Python um programa que leia uma temperatura em graus Fahrenheit e escreva a correspondente em graus Celsius. A fórmula utilizada para a conversão de graus Fahrenheit em graus Celsius é:

$$C = (F - 32) / 1.8$$

2. Escreva em linguagem Python um programa que leia o ano de nascimento de uma pessoa e escreva no écran a idade que terá no final do ano actual. Segue-se um exemplo da interação com o computador.

```
Indique o ano de nascimento: 1972  
No final de 2015 terá 42 ano(s).
```

3. Escreva em linguagem Python um programa que leia um intervalo de tempo em horas, minutos e segundos, e que depois imprima o número de segundos equivalente. Por exemplo, 1 hora, 28 minutos e 42 segundos é equivalente a 5322 segundos.
4. Desenvolva a versão inversa do problema anterior: lê um número representando uma duração em segundos e imprime o seu equivalente em horas, minutos e segundos. Por exemplo, 9999 segundos é equivalente a 2 horas, 46 minutos e 39 segundos
5. Considere este excerto de código em Python:

```
if x == 1:  
    x = x + 1  
    if x == 1:  
        x = x + 1  
    else:  
        x = x - 1  
else:  
    x = x - 1
```

- a. Se no início da execução a variável x tiver o valor 1, qual o valor de x no final da execução?
 - b. Qual teria de ser o valor inicial de x para que no final da execução fosse -1?
 - c. Independentemente do valor inicial de x, há uma parte deste programa que nunca é executada: qual e porquê?
6. Escreva em linguagem Python um programa que leia um número inteiro positivo (menor ou igual a 5) e escreva no ecrã a sua representação em numeração romana. Segue-se um exemplo da interacção com o computador (em itálico os dados introduzidos pelo utilizador).

```
Diga um numero: 4
IV
```

7. Escreva em linguagem Python um programa que leia um ano (> 0) e escreva no ecrã o século a que este ano pertence. Relembre que o ano 1999 faz parte do século XX, 2000 ainda faz parte do século XX e que 2001 é o primeiro ano do século XXI. Desenvolva uma versão do programa utilizando um comando alternativo e outra utilizando apenas uma expressão numérica. Note que `int(True)=1` e que `int(False)=0`.
8. Escreva em linguagem Python um programa que leia uma data (ano, mês e dia em separado) e um número inteiro x, e devolva uma nova data x dias mais à frente.
9. Indique os erros sintáticos no seguinte programa em Python:

```
x = 1
y = 1
while x = 1 and y < 5
    y = y + 2
```

10. Este programa em Python tem como objectivo escrever a tabuada do número inteiro dado pelo utilizador. Explique porque é que este programa não termina, corrija o erro e especifique qual o valor das variáveis n e i no final da execução do programa corrigido:

```
n = int(input("Escreve um número inteiro: "))
print("Tabuada do", n, ":")
i = 1
while i <= 10:
    print(n, "x", i, "=", n * i)
    i + 1
```

11. Considere este programa em Python:

```

dividendo = int(input("Dividendo: "))
divisor = int(input("Divisor: "))
resto = dividendo
quociente = 0
while resto >= divisor:
    resto = resto - divisor
    quociente = quociente + 1
    print("O quociente é", quociente, "e o resto é", resto)

```

- a. O que faz este programa?
 - b. Quais são as duas operações aritméticas disponíveis em Python que implementam a mesma funcionalidade?
 - c. Altere o programa de forma a que, caso o dividendo seja menor que o divisor, o utilizador seja alertado e lhe sejam pedidos novos valores.
 - d. Faça uma alteração adicional de forma a garantir também que tanto o dividendo como o divisor são positivos.
12. O seguinte programa em Python escreve no ecrã os factoriais de todos os números inteiros entre 1 e n, em que n é dado pelo utilizador:

```

n = int(input("Escreve um número inteiro: "))
current_n = 1
while current_n <= n:
    i = current_n
    f = 1
    while i > 1:
        f = f * i
        i = i - 1
    print("Factorial de " + str(current_n) + ": " + str(f))
    current_n = current_n + 1

```

Altere o programa de forma a que o programa não escreva os factoriais cujo valor é superior a 1000. (Note que há formas mais ou menos eficientes de o fazer.)

13. Escreva um programa em Python que peça ao utilizador um número decimal e escreva no ecrã a sua parte inteira perguntando em seguida se o utilizador quer terminar a utilização do programa.
14. Escreva um programa em Python que peça ao utilizador um número inteiro k e uma palavra w e escreva no ecrã k linhas em que cada linha i tem o número da linha i separado por um espaço da palavra w concatenada i vezes, com i no intervalo de 1 a k.

15. Escreva um programa em Python que peça ao utilizador um número inteiro k e escreva no ecrã o resultado do produto das potências de 3 com expoente de 0 a k .
16. Escreva um programa em Python que peça ao utilizador um número inteiro k maior do que 2 e escreva no ecrã quantos números perfeitos existem entre 2 e k (inclusive). Por exemplo, existem 4 números perfeitos entre 2 e 10000 (o 6, o 28, o 496 e o 8128).
17. Escreva um programa em Python que peça ao utilizador um número inteiro k maior do que 10 e escreva no ecrã quantas capícuas existem entre 10 e k . Uma capícua é um número que se lê de igual forma da esquerda para a direita e da direita para a esquerda. Por exemplo, entre 10 e 100 existem 9 capícuas