

# Exercícios do Curso de Python

## Ficha 2

### Funções simples

1. Considere a seguinte função:

```
def imprimeDivisaoInteira(x, y):  
    if y == 0:  
        print "Divisao por zero"  
    else:  
        print x//y
```

- O que faz esta função?
- Qual o resultado da seguinte sequência de comandos?

```
imprimeDivisaoInteira(4, 2)  
imprimeDivisaoInteira(2, 4)  
imprimeDivisaoInteira(3, 0)  
help(imprimeDivisaoInteira)  
imprimeDivisaoInteira()
```
- Altere a definição da função de modo a adoptar a abordagem da programação por contratos (recorrendo a uma docstring) em vez da abordagem da programação defensiva.

2. Considere a seguinte função:

```
def potencia(a, b):  
    return a**b
```

- O que faz a seguinte sequência de comandos?

```

a = 2
b = 3
potencia(b, a)
potencia(a, b)
print potencia(b, a)
print potencia(a, b)
print potencia(2,0)
print potencia(2)

```

- b. Escreva uma nova função potenciaP que receba apenas um número inteiro a e retorne  $a^a$ .
  - c. Escreva uma nova versão da função potencia que possa ser chamada com dois ou com um argumento e, consoante o caso, execute  $a^b$  ou  $a^a$ .
3. Considere o seguinte programa:

```

a = 4
def printFuncao():
    a = 17
    print("Dentro da funcao: ", a)
printFuncao()
print("Fora da funcao: ", a)

```

- a. Qual é o resultado de executar este programa?
- b. O que é uma variável global? E uma variável local?

4. Considere o seguinte código:

```

def somaDivisores(num):
    """
    Soma de divisores de um numero dado

    Requires:
        num seja int e num > 0
    Ensures: um int correspondente à soma dos divisores
    de num que sejam maiores que 1 e menores que num
    """

```

- a. Como cliente da função somaDivisores, o que deve satisfazer para a função cumprir o contrato?
  - b. E o que obtém da função se a chamar satisfazendo a sua pré-condição?
5. Crie um programa que pergunte sucessivamente ao utilizador um número inteiro positivo e imprima a soma dos seus divisores. A execução do programa deve terminar quando o utilizador introduzir um número negativo. Nota: assuma que a função somaDivisores, apresentada no exercício anterior, se encontra definida.

6. Considere o seguinte programa:

```
DIA_ATUAL = 2
MES_ATUAL = 11
ANO_ATUAL = 2015

print "Dados do Pai"
anoPai = int(raw_input("Introduza o ano de nascimento: "))
mesPai = int(raw_input("Introduza o mes de nascimento: "))
diaPai = int(raw_input("Introduza o dia de nascimento: "))

print "Dados da Mae"
anoMae = int(raw_input("Introduza o ano de nascimento: "))
mesMae = int(raw_input("Introduza o mes de nascimento: "))
diaMae = int(raw_input("Introduza o dia de nascimento: "))

if mesPai > MES_ATUAL or \
    (mesPai == MES_ATUAL and diaPai > DIA_ATUAL):
    print("Pai tem", ANO_ATUAL - anoPai - 1, "ano(s)")
else:
    print("Pai tem", ANO_ATUAL - anoPai, "ano(s)")

if mesMae > MES_ATUAL or \
    (mesMae == MES_ATUAL and diaMae > DIA_ATUAL):
    print("Mae tem", ANO_ATUAL - anoMae - 1, "ano(s)")
else:
    print("Mae tem", ANO_ATUAL - anoMae, "ano(s)")
```

Recorrendo a funções, simplifique o programa apresentado eliminando a repetição de código.

7. Escreva uma função que receba dois números inteiros e devolva o maior deles. Inclua o contrato da função. Teste a função escrevendo um programa que receba dois números inteiros do utilizador e imprima o resultado da chamada à função desenvolvida. Como teria de fazer para determinar o menor de dois números com uma segunda função que tirasse partido de chamar a primeira?
8. Escreva uma função (o contrato não pode ser esquecido) que elimine a casa das unidades de um número inteiro. Por exemplo, `retira(537)` devolve 53. Se o argumento só tiver algarismo das unidades, a função deve devolver 0. Teste a função escrevendo um programa que receba um número inteiro do utilizador e imprima o resultado de chamada à função desenvolvida.
9. Escreva uma função que acrescente um 0 no final de um número inteiro. Por exemplo, `aumenta(73)` devolve 730 (se esquecer o contrato a definição da função está incompleta). `aumenta(0)` deve devolver 0. Teste a função escrevendo um programa que receba um número inteiro do utilizador e imprima o resultado da chamada à função desenvolvida.

10. Escreva uma função que receba um número inteiro e devolva a soma dos divisores próprios desse número

```
def somaDivisores(num):  
    """  
    Soma dos divisores próprios de um numero dado.  
    Requires:  
        num seja int e num > 0  
    Ensures:  
        um int correspondente à soma dos divisores  
        de num que sejam maiores que 1 e menores que num  
    """
```

Teste a função escrevendo um programa que receba um número inteiro do utilizador e imprima o resultado da chamada à função desenvolvida.

11. Escreva uma função que verifique se um dado número dado é primo. Relembre que um número é primo se é maior do que 1 e não tem divisores próprios. Teste a função escrevendo um programa que receba um número inteiro do utilizador e imprima o resultado da chamada à função desenvolvida.
12. Usando a função do exercício anterior, escreva um programa que receba um número inteiro  $n$  maior do que 2 e escreva no ecrã quantos números primos existem entre 2 e  $n$  (inclusive). Por exemplo, existem 1000 números primos entre 2 e 7919. Teste a função escrevendo um programa que receba um número inteiro do utilizador e imprima o resultado de chamada à função desenvolvida. Explique, nesta situação, em que difere a programação defensiva da programação por contratos.