

Programação em Python

Tópicos

- Funções: definição, parâmetros, variáveis locais

Exercícios

1. Escreva uma função para calcular o polinómio $p(x)=x^2+2x+3$ e use-a num programa para calcular e mostrar os valores de $p(0)$, $p(1)$, $p(2)$ e $p(p(1))$. Confira os resultados.

Proposta resolução:

```
# Pensamento Computacional
# Aula 5
# Exercício 1
#
def pol(x):
    pol=pow(x,2)+2*x+3
    return pol

print("p(0)=",pol(0))
print("p(1)=",pol(1))
print("p(2)=",pol(2))
aux=pol(1)
print("p(p(1))=",pol(aux))
```

2. Defina uma função que devolva o maior dos seus dois argumentos. Por exemplo, $\text{max2}(5, 3)$ deve devolver 5 enquanto $\text{max2}(-3, -2)$ deve devolver -2. Não pode usar a função pré-definida `max`. Use uma instrução de seleção `if` ou uma expressão condicional. Teste a função com vários conjuntos de argumentos.

Proposta resolução:

```
#
# Exercício 2
#
def max2(x,y):
    if x<y: return y
    else: return x

print("(5,3)", max2(5,3))
print("(-3,-2)", max2(-3,-2))
```

3. Escreva uma função, $\text{tax}(r)$, que implemente a seguinte função de ramos:

$$\text{tax}(r) = \begin{cases} 0.1r & \text{se } r \leq 1000 \\ 0.2r - 100 & \text{se } 1000 < r \leq 2000 \\ 0.3r - 300 & \text{se } 2000 < r \end{cases}$$

Use uma instrução `if-elif-else` e evite condições redundantes. Teste a função para diversos valores de r e confirme os resultados. Que valores deve testar?

Proposta resolução:

```
#  
# Exercício 3  
#  
def tax(r):  
    if r<=1000: tax=0.1*r  
    elif r<=2000: tax= 0.2*r-100  
    else: tax=0.3*r-300  
    return tax  
print("r=1000 ", tax(1000))  
print("r=1500 ", tax(1500))  
print("r=2000 ", tax(2000))  
print("r=2100 ", tax(2100))
```

4. Escreva a função `hms2sec` que deve devolver o número de segundos correspondente a h horas, m minutos e s segundos. Teste a função com vários conjuntos de argumentos.

Proposta resolução:

```
#  
# Exercício 4  
#  
def hms2sec(h, m, s):  
    sec = (h*60 + m) * 60 + s  
    return sec  
print("1,2,3", hms2sec(1,2,3))
```

5. ** Escreva uma função `countdown(N)` que imprima uma contagem decrescente a partir de um número positivo N . Note que pode imprimir N e depois fazer `countdown(N-1)`. Teste a função com diversos valores de N .

Proposta resolução:

```
#  
# Exercício 5  
#  
def countdown(n):  
    if n>=1:  
        print(n)  
        countdown(n-1)  
countdown(10)
```