

FUNÇÃO E ESTRUTURAS DE REPETIÇÃO EM PYTHON

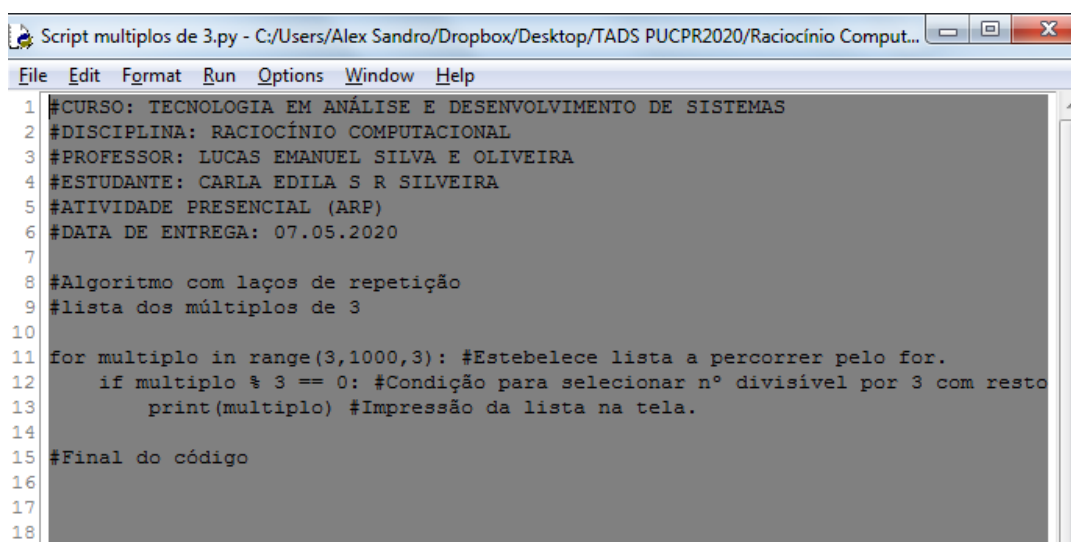
Carla Edila Santos da Rosa Silveira (PUCPR)

E-mail: rosa.carla@pucpr.edu.br

É possível explicar a utilidade de cada elemento que compõe uma função em Python a partir do código exemplificado no enunciado da questão 1. Na primeira linha do código (no cabeçalho) há 4 elementos. Temos como primeiro deles a palavra reservada “**def**” que é utilizada para criar uma função, ou seja, uma parte do código que executa determinada tarefa. O “def” é considerado palavra reservada por seu uso exclusivo para **definição da função** na programação em Python. O **nome** da função é o segundo elemento e, no exemplo, corresponde a “multiplicar_numeros”. Para utilizar a função em outra linha do programa é preciso chamá-la pelo nome declarado no seu cabeçalho. Como terceiro elemento, encontramos os **parâmetros de entrada** da função entre parênteses; no exemplo é a parte “(a,b)”. Os parâmetros de entrada indicam que a função receberá dados através de argumentos ou variáveis representados pelas letras “a” e “b”. Por último vêm **dois pontos** para encerrar a linha do cabeçalho. Já a segunda linha inicia com **indentação** (adentramento de 4 espaços ou usar uma vez a tecla Tab do teclado) para incluir **comando** (return) que realizará a tarefa de atribuída à função: multiplicação (com asterisco como operador) das variáveis (“a*b”).

A principal vantagem de uma função em Python consiste na organização do código em subpartes com possibilidade de reutilização quando seu nome é chamado em outro ponto do programa. O comando “return” habilita a função para uso posterior. Evitar repetições de mesmas linhas de instruções é outra vantagem que torna o código limpo. A criação de funções pode apresentar dificuldades se houver falha na inclusão de elementos nos escopos: global no início do programa (onde a função será definida) e local (onde a função será chamada para receber valores). O trecho com a função não executa nada isoladamente, pois esta precisa ser chamada em outra linha do código para executar sua tarefa com as variáveis que entrarem no escopo local.

Os comandos “while” e “for” possibilitam a criação de estruturas de repetição ou laços nos códigos em Python, evitando repetição de mesmas linhas no código. Os dois comandos podem executar as mesmas tarefas, mas a sintaxe do “while” é diferente da usada com “for”. Com a estrutura de repetição alteramos o fluxo linear de execução do programa e é por meio de variáveis contadoras e acumuladoras que controlamos o laço (ou loop). A repetição feita com “while” ocorre enquanto a condição for verdadeira. Com o “for” a estrutura de repetição é feita com base em listas; podemos fazer contagens segundo critérios de números (pares, ímpares, múltiplos) e operações matemáticas como somatória, fatorial e progressões (PA e PG).



```
Script multiplos de 3.py - C:/Users/Alex Sandro/Dropbox/Desktop/TADS PUCPR2020/Raciocinio Comput...
File Edit Format Run Options Window Help
1 #CURSO: TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
2 #DISCIPLINA: RACIOCÍNIO COMPUTACIONAL
3 #PROFESSOR: LUCAS EMANUEL SILVA E OLIVEIRA
4 #ESTUDANTE: CARLA EDILA S R SILVEIRA
5 #ATIVIDADE PRESENCIAL (ARP)
6 #DATA DE ENTREGA: 07.05.2020
7
8 #Algoritmo com laços de repetição
9 #lista dos múltiplos de 3
10
11 for multiplo in range(3,1000,3): #Estabelece lista a percorrer pelo for.
12     if multiplo % 3 == 0: #Condição para selecionar n° divisível por 3 com resto
13         print(multiplo) #Impressão da lista na tela.
14
15 #Final do código
16
17
18
```