

Subprogramas em Python

Introdução à Programação para Biologia Molecular

Rosane Minghim

Apoio na confecção: Danilo Medeiros Eler

Rogério Eduardo Garcia

Renato Rodrigues

Carlos E. A. Zampieri

Baseado na Apostila: Curso Introductório de Computação por R. Minghim e G. P. Telles

Subprogramas

- Subdivisão de um problema em problemas menores:
 - Desenvolvimento de cada etapa individualmente;
- Clareza e Compacidade:
 - Código mais curto e fácil de ler;
- Generalização e Aproveitamento de código:
 - Utilização de algoritmos genéricos.

Subprogramas em Python

#NOME

Relação das Unidades Utilizadas
Definição de Constantes

Definição de Funções e Procedimentos

Comandos

Subprogramas em Python

#NOME

Relação das Unidades Utilizadas
Definição de Constantes

Definição de Funções e Procedimentos

Comandos

A definição de uma função, ou procedimento, deve estar **antes** de sua invocação

Notação - Subprogramas

FUNÇÕES

Subprograma nome (parâmetro, parâmetro, ..., parâmetro):**tipo**

e: parâmetro: tipo
parâmetro: tipo
...

s: parâmetro: tipo
...

e/s: parâmetro: tipo
...

r: tipo

declarações

início
comandos do algoritmo do subprograma
retorne(expressão)
fim

Exemplo de Algoritmo

Subprograma média_aritmética (valor1, valor2, valor3):real

e: valor1, valor2, valor3: real
{valores dos quais se deseja calcular a média aritmética}
r: média aritmética dos três parâmetros de entrada, real.

início
retorne((valor1+valor2+valor3)/3)
fim

Em Python

```
def NOME-DA-FUNÇÃO (parâmetro1, parâmetro2,  
...  
parâmetron):  
  
    corpo da função
```

Em Python

```
def NOME-DA-FUNÇÃO (parâmetro1, parâmetro2,  
...  
parâmetron):  
  
    corpo da função  
    return valor
```

Valem as mesmas regras para nomes de variáveis

Palavra reservada

Variáveis utilizadas no **corpo da função** depois de terminada a função elas deixam de existir

Em Python

```
def NOME-DA-FUNÇÃO (parâmetro1, parâmetro2,  
...  
parâmetron):
```

corpo da função
return valor

Palavra reservada que define o retorno de um valor da função

PARÂMETRO FORMAL
quando a função
for utilizada, este parâmetro
recebe um valor real
lido ou atribuído no
programa principal

Em Python

```
def media_aritmetica(valor1, valor2, valor3):
```

```
    media = (valor1+valor2+valor3)/3  
    return media
```

O valor de retorno deve ser atribuído após a palavra reservada **return**.

Em Python

```
def media_aritmetica(valor1, valor2, valor3):
```

```
    soma = (valor1+valor2+valor3)  
    media = soma/3  
    return soma, media
```

Variáveis na ordem de retorno do subprograma.

```
soma, media = media_aritmetica(6, 7, 8)  
print soma, media
```

Saída: "21 7"

Pode-se retornar vários valores com a cláusula **return** separando-os por vírgulas na ordem desejada de retorno.

Exemplo

```
#exemplo_de_uso_de_funcoes;  
#Nro: integer - número para calcular o fatorial  
#resultado: real - resultado do fatorial de N  
  
def FAT(nro):  
    result = 1.0  
    for auxiliar in range(1,nro):  
        result = result * auxiliar  
    return result  
  
Nro = input('FORNECA O VALOR DE N: ')  
resultado = FAT(Nro)  
print 'O fatorial de {0:2d} eh {1:.2f}'.format(Nro, resultado)
```

Notação - Subprogramas

PROCEDIMENTOS:

Subprograma nome (parâmetro, parâmetro, ..., parâmetro)

e: parâmetro: tipo
parâmetro: tipo
parâmetro: tipo
...

s: parâmetro: tipo
parâmetro: tipo
parâmetro: tipo
...

e/s: parâmetro: tipo
parâmetro: tipo
parâmetro: tipo
...

declarações

início
comandos do algoritmo do subprograma
fim

Em Python

```
def NOME-DO-PROCEDIMENTO (parâmetro1, parâmetro2,  
...  
parâmetron):  
  
    corpo do procedimento
```

Em Python

Palavra
reservada

Valem as mesmas regras para
nomes de variáveis

```
def NOME-DO-PROCEDIMENTO (parâmetro1, parâmetro2,  
...  
parâmetron):  
  
    corpo do procedimento
```

Variáveis utilizadas no **corpo do
procedimento** depois de terminado o
procedimento elas deixam de existir

Em Python

PARÂMETRO FORMAL
quando a função
for utilizada, este parâmetro
recebe um valor real
lido ou atribuído no
programa principal.

```
def NOME-DO-PROCEDIMENTO (parâmetro1, parâmetro2,  
...  
parâmetron):  
  
    corpo do procedimento
```

Parâmetros

- Parâmetros de entrada são declarados como visto até aqui
 - PASSAGEM DE PARÂMETRO POR VALOR
- Parâmetros de saída e entrada/saída não existem em python
 - PASSAGEM DE PARÂMETRO POR REFERÊNCIA

Em Python

Passagem de parâmetro por valor

```
def NOME-DO-PROCEDIMENTO (parâmetro1, parâmetro2,  
    ...  
    parâmetron):  
  
    corpo do procedimento
```

Passagem de Parâmetros

- Por Valor:
 - Uma cópia do valor do parâmetro REAL é feita para o parâmetro FORMAL.

```
def KKK(parametro1):  
    corpo  
    ...  
    KKK(numero)
```

Supondo que a variável *numero* armazene o valor 10, este é copiado para *parametro1* do procedimento KKK

Passagem de Parâmetros

- Por Referência:
 - Uma forma de resolver a passagem de parâmetro por referência em Python é utilizar uma **Função** para retornar o valor alterado.

```
def KKK(parametro1):  
    parametro1 = 20  
    return parametro1  
numero = KKK(numero)
```

Supondo que a variável *numero* armazene o valor 10, depois de executado o procedimento KKK, seu valor será 20.

Escopo – Variável Global

- Uma **variável global** é uma variável definida na seção de declarações de um programa principal, e ela pode ser utilizada em qualquer lugar no arquivo do programa principal, isto é, dentro do próprio programa ou de qualquer subprograma definido internamente a ele.

Escopo – Variável Local

- Uma **variável local**, isto é, declarada dentro de um subprograma, tem seu escopo restrito a este subprograma, isto é, só pode ser utilizada no bloco de comandos do subprograma e nos outros subprogramas definidos dentro dele

Escopo – Definição Dupla

- Se uma variável é duplamente definida, vale o escopo mais local

Evitar a todo custo

- O uso de variáveis globais dentro de subprogramas (ou variáveis fora do escopo do próprio subprograma) é uma prática extremamente nociva aos princípios de boa programação

Exercícios Sugeridos

- 1) Dados dois números N e K, calcular a combinação

$$C_{N,K} = \frac{N!}{K!(N-K)!}$$

- 2) Escrever um procedimento que, dados três números, obtenha o maior e o menor deles. Faça também o programa principal para testá-lo.

Subprogramas em Python

Introdução à Programação em Python para Biologia Molecular

Rosane Minghim

Apoio Técnico: Danilo Medeiros Eler

Rogério Eduardo Garcia

Renato Rodrigues

Carlos E. A. Zampieri

Baseado na Apostila: Curso Introductório de Computação por R. Minghim e G. P. Telles