Programación Numérica para Geofísica

PNG 2020-1

Andrés Sepúlveda

Departamento de Geofísica
Universidad de Concepción

01 Junio 2020

Anuncios

- Hoy: **Funciones**

- **Funciones:** conjunto de código que realiza una operación sobre ciertos argumentos de entrada.
- Generalmente entrega algo como salida, ya sea un numero, una matriz, o un gráfico.
- Se pueden definir dentro del archivo principal, o externamente.
- Permite reutilizar un código y encontrar posibles errores.
- Si calculan algo de forma repetida, ¡hagan una función!
- Desafío mayor: anticipar todas las formas en que el usuario pueden equivocarse al ingresar los argumentos.

Octave

Funciones

- Comienzan con *function*
- Después el nombre de la salida de la función,
- luego el nombre de la función,
- y los argumentos entre paréntesis.
- No se indentan las líneas.
- No olviden los ; en cada línea.
- Finaliza con un **end**.
- ¡Agreguen comentarios a su código!

```
function area =
     areacirculo(radio)
%
%  Area de un circulo
%  Input: radio  (en km)
%  Output: area (en km **2)
%  DGEO-UdeC 01/06/2020
%
 area = 3.14 * (radio**2);
end

a = areacirculo(3)
```

Octave

Funciones

- Si queremos varios argumentos de salida, definimos la función de la siguiente forma:
- El nombre del archivo que contiene la función debe ser igual al nombre de la función, en este caso el archivo debe llamarse **dolittle.m**

```
function [out1,out2] =
        dolittle (x)
   out1 = x^2;
   out2 = out1*x;
 end
```

Octave/Matlab

IMPORTANTE

- Al llamar una función hay que asignar valores a todos los argumentos de esta.
- Una función bien diseñada asigna valores por defecto si el usuario no los escribe. Para esto se usa la función **nargin**
- Una limitación de Matlab/Octave es que hay que agregar los valor es de los argumentos en orden. Python evita eso al usar *diccionarios*.

```
function area = area_elipse(rad1,rad2,pi)
if nargin < 3
pi = 3;  % Valor malo a proposito!
end
radprom = (rad1 + rad2)/2;
area = pi * (radprom*radprom);
end
```

```
a = area_elipse(3,1)
a = area_elipse(3,1,3.1416)
a = area_elipse(3,,3.11425)  % No funciona!
```

Octave/Matlab

IMPORTANTE

- Una función bien diseñada verifica los argumentos.

```
function retval = avg (v)
  retval = 0;
  if (length(v) > 1)
    retval = sum (v) / length (v);
  else
    error ("avg: argumento debe ser un vector");
  endif
end
```

Octave/Matlab

IMPORTANTE

- Una función bien diseñada verifica los argumentos.

```
function retval = avg (v)
  retval = 0;
  if (nargin != 1)
    usage ("avg (vector)");
  endif
  if (length(v) > 1)
    retval = sum (v) / length (v);
  else
    error ("avg: argumento debe ser un vector");
  endif
endfunction
```

Octave/Matlab

IMPORTANTE

- Una función bien diseñada verifica los argumentos.

try/catch

```
try
    b=max(v);
catch
  disp('El argumento v no es un número')
end_try_catch
```

Funciones usadas

- ¡Uds. ya han usado funciones en Octave!
- Busquen alguna función (*which*), y estudien el código de esta.
- Escriban una breve función.