



# Informática

## Curso 2023

Facultad de Ingeniería  
Universidad de la República

# Estructuras de control

- Nivel de instrucción

- Especifican flujo de control entre instrucciones de programa.

- Secuencia

- Selección

- Iteración

- Nivel de Unidad

- Especifican flujo de control entre unidades de programa.

- Subprogramas 

# Estructuras de control

## Subprogramas

- Un enfoque para la resolución de un problema complejo es el de subdividirlo en problemas más simples y luego resolver estos últimos.
- Permiten agrupar código de una forma lógica.
- Octave
  - Scripts
  - Funciones

# Estructuras de control

## Subprogramas

### ■ *Invocación*

- Un procedimiento simple es invocado escribiendo su nombre como una sentencia ejecutable.
- Cuando un procedimiento es invocado:
  - La dirección de la sentencia siguiente a la invocación es memorizada.
  - Se ejecutan las sentencias del procedimiento.
  - La ejecución retorna al punto de invocación

# Estructuras de control

## Subprogramas

- Un subprograma puede ser usado cuando una secuencia de sentencias aparece repetidamente en un programa.
- Agrupar sentencias sugiere un propósito particular de las mismas.
- Programas mantenibles
- Cada uno de los procedimientos en un programa puede ser escrito por distintos programadores.

# Octave

## Archivos .m

- Ejecución de instrucciones almacenadas en archivos de disco.
- Se llaman archivos .m pues su sufijo debe ser m
- Dos tipos
  - Archivos de instrucciones o scripts
  - Archivos de funciones

# Octave

## Scripts

- Sucesión de comandos o instrucciones guardados en un archivo .m
- Ejecución
  - Para invocarlo se lo llama por el nombre del archivo.
  - Se ejecuta como si se estuvieran digitando los comandos en la consola, uno tras otro.
  - Pueden ser invocados desde el entorno o desde otro script.

# Octave

## Scripts

### ■ Variables

- Alcance **global**.
- Puede utilizar y modificar las variables definidas en el entorno interactivo.
- Puede definir nuevas variables en el entorno interactivo.

### ■ Utilidad

- Programas utilizados repetidamente.
- Inicialización de variables de entorno.



# Octave

## Scripts

- Archivo mi\_script.m
- Contenido

```
x = 10;  
b = x + 4;  
z = x + b + 3;
```

- Ejecución
  - >> mi\_script
- Resultado
  - Se crean las variables x, b, y z en el entorno interactivo.
  - Sus valores serán
    - x = 10
    - b = 14
    - z = 27

# Octave

## Funciones

- Sucesión de comandos o instrucciones guardados en un archivo .m
- Posee características propias de las funciones matemáticas
  - Recibe valores (parámetros) de entrada.
  - Realiza algún cálculo o tarea.
  - Devuelven un resultado (parámetros de salida)

# Octave

## Funciones

### ■ Sintaxis

- Cabezal del archivo .m

```
function [salida1, salida2,...] = nombre_funcion  
(ent1, ent2, ...)
```

- Las *salida<sub>i</sub>* son el resultado que devuelve la función
  - Pueden ser un vector o una matriz
- Los parámetros *ent<sub>i</sub>* son los datos necesarios para la ejecución de la función.

# Octave

## Funciones

### ■ Sintaxis

- Al final del archivo

**endfunction**

- Indica el final de la función.

# Octave

## Funciones

### ■ Ejecución

- El nombre de la función **puede** coincidir con el nombre del archivo `.m` y es **recomendable** que así sea.
- Se invoca por su nombre indicando los parámetros necesarios para su ejecución.
- Pueden ser invocadas desde la consola (o script) o desde otra función.

# Octave

## Funciones

### ■ Variables

- Alcance **local**, crea y encapsula sus propias variables.
- Estas variables **existen** el tiempo que dure la **ejecución** de la función.
- **NO** afecta variables definidas en el entorno interactivo.
- **NO** puede definir nuevas variables en el entorno interactivo.

# Octave

## Funciones

- Archivo mi\_funcion.m
- Contenido

```
function z = mi_funcion(x)
    b = x + 4;
    z = x + b + 3;
endfunction
```

- Ejecución
  - >> mi\_funcion(10)
- Resultado
  - No se crea ninguna variable en el entorno interactivo.

# Octave Funciones

File Edit Debug Window Help News

Current Directory: C:\Users\Martin\octave

File Browser

C:/Users/Martin/octave

Nombre

- mi\_funcion.m
- mi\_script.m

Workspace

Name	Class	Dimension	Value	Attribute
ans	double	1x1	27	

Command History

- x
- b
- z

Command Window

GNU Octave, version 4.0.3  
Copyright (C) 2016 John W. Eaton and others.  
This is free software; see the source code for copying conditions.  
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or  
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.  
  
Octave was configured for "i686-w64-mingw32".  
  
Additional information about Octave is available at <http://www.octave.org>.  
  
Please contribute if you find this software useful.  
For more information, visit <http://www.octave.org/get-involved.html>.  
  
Read <http://www.octave.org/bugs.html> to learn how to submit bug reports.  
For information about changes from previous versions, type 'news'.  
  
>> mi\_funcion(10)  
ans = 27  
>> x  
error: 'x' undefined near line 1 column 1  
>> b  
error: 'b' undefined near line 1 column 1  
>> z  
error: 'z' undefined near line 1 column 1  
>> |

Command Window Editor Documentation

Windows taskbar: 13:17 24/08/2016



# Octave

## Funciones

- **No utilizar** instrucciones para desplegar resultados dentro de una función.
  - Los resultados de una función se deben devolver en los parámetros de salida.
- **No utilizar** instrucciones para leer los datos de entrada dentro de una función.
  - Las entradas de una función se deben cargar en los parámetros de entrada.

# Pregunta interesante

## Selección

- ¿Qué es mejor usar ifs en secuencia o if-elseif?

```
if (calificación >= 25) and (calificación < 60)
    nota = 4; % Curso Aprobado
endif
if (calificación < 25)
    nota = 0; % Curso Reprobado
endif
if (calificación >= 60)
    nota = 6; % Exoneración
endif
```

# Pregunta interesante

## Selección

- ¿Qué es mejor usar ifs en secuencia o if-elseif?

```
if (calificación >= 25)and(calificación < 60)
    nota = 4;% Curso Aprobado
elseif (calificación < 25)
    nota = 0;% Curso Reprobado
elseif (calificación >= 60)
    nota = 6; % Exoneración
endif
```

# Pregunta interesante

## Selección

- ¿Qué es mejor usar ifs en secuencia o if-elseif?
- El if-elseif es más expresivo que los ifs en secuencia.
- Usando if-elseif el código es **más claro** ya que se aprecia fácilmente que los casos son excluyentes, mientras que usando ifs hay que revisar caso a caso.

# Pregunta interesante

## Iteración - For

- ¿Debo usar obligatoriamente el contador en el cuerpo del for?
- El contador permite distinguir que paso de la repetición se está procesando.
- Dependerá del problema si el contador resulta útil, es decir **si uso su valor en el cuerpo o no.**