



# Fundamentos de Informática

## Estructuras Condicionales

### Dept. Lenguajes y Sistemas Informáticos (LSI)

#### 2018/2019

Rodrigo Agerri, Xabier Larrucea, Mari Carmen Otero, Juan Francisco Ramirez



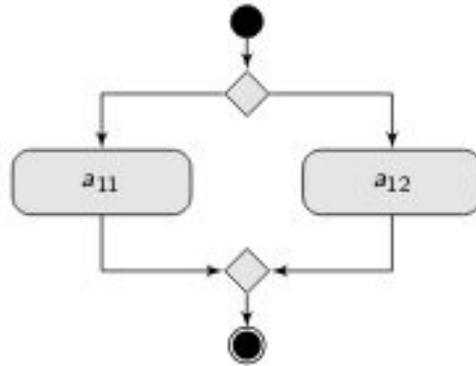
# Índice



1. Introducción
2. Expresiones condicionales
3. Estructuras "if"
4. Estructuras "switch"
5. Funciones lógicas

# Motivación

- Las estructuras condicionales son necesarias para realizar ciertas operaciones, por ejemplo:
  - Mostrar los números negativos en un vector



# Operadores relacionales

Se utilizan para comparar valores y devuelven un valor lógico: verdadero o falso

>	Mayor
<	Menor
>=	Mayor o igual
<=	Menor o igual
==	Igual
~=	No es igual

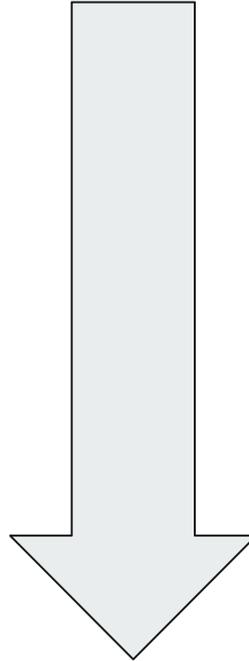
# Operadores lógicos

Permiten concatenar operadores relacionales u otros operadores lógicos

	Verdadero si una de las dos condiciones se cumple (disyunción lógica inclusiva)
&&	Verdadero si las dos condiciones se cumplen (conjunción lógica)
~	Negación lógica

# Prioridad de operadores

()
^
- (negativo) ~
* /
+ - (resta)
< <= > >= == ~=
&&
=



# if



```
if <condición>  
  enunciados  
end
```

```
if num < 0  
  num = 0;  
end
```

# if-else



```
if <condición>  
    enunciados
```

```
else
```

```
    enunciados  
end
```

```
if resultado == 7  
    disp('correcto');
```

```
else
```

```
    disp('incorrecto');  
end
```

## if-else-if



if <condición-1>  
    enunciados

elseif <condición-2>  
    enunciados

else  
    enunciados  
end

**if** resultado == 7  
    disp('correcto');

**elseif** resultado < 7  
    disp('mayor');

**else**  
    disp('menor');  
**end**

# if anidados



```
if x >= 0
    if x < 4
        disp('a')
    else
        disp('b')
    end
else
    disp('c')
end
```

# Ejemplo



```
a = 100;
%check the boolean condition
if a == 10
    % if condition is true then print the following
    fprintf('Value of a is 10\n' );
elseif( a == 20 )
    % if else if condition is true
    fprintf('Value of a is 20\n' );
elseif a == 30
    % if else if condition is true
    fprintf('Value of a is 30\n' );
else
    % if none of the conditions is true '
    fprintf('None of the values are matching\n');
fprintf('Exact value of a is: %d\n', a );
end
```

## Ejemplo (ii)



Determine si un valor está dentro de un rango especificado.

```
x = 10;
minVal = 2;
maxVal = 6;

if (x >= minVal) && (x <= maxVal)
    disp('Value within specified range.')
elseif (x > maxVal)
    disp('Value exceeds maximum value.')
else
    disp('Value is below minimum value.')
end
```

Value exceeds maximum value.

<https://es.mathworks.com/help/matlab/ref/if.html>

# switch case



## Sintaxis

---

```
switch switch_expression
  case case_expression
    statements
  case case_expression
    statements
  ...
  otherwise
    statements
end
```

Evalúa una expresión y elige ejecutar uno de varios grupos de instrucciones. Cada elección es un caso.

El bloque switch prueba cada caso hasta que una de las expresiones Case sea true. Un caso es verdadero cuando:

- Para los números, *case\_expression* == *switch\_expression*.
- Para los vectores de caracteres, `strcmp(case_expression,switch_expression) == 1`.
- Para los objetos que admiten la función eq , *case\_expression* == *switch\_expression*.
- Para una matriz de celdas *case\_expression*, al menos uno de los elementos de la matriz de celdas coincide con *switch\_expression*, como se define anteriormente para los números, los vectores de caracteres y los objetos.

# Funciones: llamadas



isnumeric	evalúa si el valor es numérico; con vectores evalúa todos los elementos
isletter	Evalúa si el valor es una letra; con vectores evalúa todos los elementos
ischar	Evalúa si el valor es un vector de caracteres
isempty	Evalúa si el vector de entrada está vacío
exist	Comprobar la existencia de variables, secuencias de comandos, funciones, carpetas o clases: exist <a href="#">track.csv file</a>