

# The *switch* construct

Allow us to select a particular code block to execute base on the value of a single integer, character, or logical expression.

```
switch (switch_expr)  
    case (case_expr)  
        statements...  
    case (case_expr2)  
        statements...  
    ....  
    otherwise  
        statements  
end
```

# The *switch* construct

Ejemplo: Lea un numero entero entre 1 y 10 y determine si es par o impar

```
N = input('Entre un numero entero entre 1 y 10: ');  
  
switch (N)  
    case {1,3,5,7,9}  
        disp('SORPRESA !!! - el numero es impar');  
    case {2,4,6,8,10}  
        disp('el numero es par, QUIEN LO CREYERA!!!');  
    otherwise  
        disp('MORON!!! aprende a leer!!!');  
end
```

# The *switch* construct

The *switch\_expr* may be either numerical or string values.

At most one block code can be executed

If the *switch expression* matches more than one *case expression* , only the first one of the will be executed.

Otherwise block code is optional.

## The *switch* construct

Elabore un programa que le pregunte al usuario el tipo de figura geométrica (rectángulo, triángulo o círculo). Dependiendo de la figura pregunte al usuario los valores de las dimensiones para calcular el área y el perímetro de la figura. Muestre los valores calculados en pantalla.

```
figura = input('Entre el nombre de la figura (rectangulo, triangulo o circulo): ','s');  
% 's' le advierte al program que se van a entrar letras (no numeros)
```

```
switch lower(figura) %lower me convierte todas las letras a minusculas
```

```
  case 'rectangulo'
```

```
    lado1 = input('Dimension del lado 1: ');  
    lado2 = input('Dimension del lado 2: ');  
    area = lado1*lado2;  
    perim = 2*(lado1+lado2);  
    disp(['El area del rectangulo es: ' num2str(area)]);  
    disp(['El perimetro del rectangulo es: ' num2str(perim)]);
```

```
  case 'triangulo'
```

```
    lado1 = input('Dimension del lado 1: ');  
    lado2 = input('Dimension del lado 2: ');  
    lado3 = input('Dimension del lado 3: ');  
    perim = 2*(lado1+lado2);  
    semip = (lado1+lado2);  
    area = (semip*(semip-lado1)*(semip-lado2)*(semip-lado3))^0.5;  
    disp(['El area del triangulo es: ' num2str(area)]);  
    disp(['El perimetro del triangulo es: ' num2str(perim)]);
```

```
  case 'circulo'
```

```
    radio = input('radio del circulo: ');  
    area = pi*radio^2;  
    perim = 2*pi*radio;  
    disp(['El area del circulo es: ' num2str(area)]);  
    disp(['El perimetro del circulo es: ' num2str(perim)]);
```

```
  otherwise
```

```
    disp('entrada no valida');
```

```
end
```

# The *try/catch* construct

Designed to trap errors.

When MATLAB encounter an error while running, the program aborts.

If an error occurs in a statement of the in the *try* block, then instead of aborting, the code in the *catch* block is executed.

If no error occurs, the statements in the catch block will be skipped

```
try
    statements
catch
    statements
end
```

# The *try/catch* construct

```
A = [1 2  
     2 1  
     3 3];
```

```
B = [3 4];
```

```
try
```

```
    C = A*B
```

```
catch
```

```
    disp('las dimensiones de las matrices no coinciden')
```

```
end
```