

CÓNQUER BLOCKS

PYTHON

TEST CONDICIONALES, IF
STATEMENT Y SWITCH-CASE

REPASO CLASE ANTERIOR

1. El uso de variables y su nomenclatura
2. Strings y como trabajar con sus funciones y métodos
3. Como trabajar con números y operaciones aritméticas
4. Combinar números y strings
5. Leer valores de entrada
6. Conversión entre tipos de datos
7. Uso de los comentarios

TEST CONDICIONALES

COMPRUEBAN SI ALGO ES CIERTO O FALSO, TRUE/FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario == 'Juan'
True
>>> nombre_usuario == 'Fede'
False
>>> █
```

Asignamos a la variable nombre_usuario un string con el símbolo ` = `

TEST CONDICIONALES

COMPRUEBAN SI ALGO ES CIERTO O FALSO, TRUE/FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario == 'Juan'
True
>>> nombre_usuario == 'Fede'
False
>>> █
```

Asignamos a la variable `nombre_usuario` un string con el símbolo `' = '`

Comparamos si la variable `nombre_usuario` es **IGUAL** al string *Juan* usando un `' == '`

TEST CONDICIONALES

COMPRUEBAN SI ALGO ES CIERTO O FALSO, TRUE/FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario == 'Juan'
True
>>> nombre_usuario == 'Fede'
False
>>> █
```

Asignamos a la variable `nombre_usuario` un string con el símbolo `` = ``

Comparamos si la variable `nombre_usuario` es **IGUAL** al string *Juan* usando un `` == ``

Efectivamente `nombre_usuario` es igual a *Juan* luego por tanto esa comparación es VERDADERA o TRUE

TEST CONDICIONALES

COMPRUEBAN SI ALGO ES CIERTO O FALSO, TRUE/FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario == 'Juan'
True
>>> nombre_usuario == 'Fede'
False
>>> 
```

Asignamos a la variable `nombre_usuario` un string con el símbolo `' = '`

Comparamos si la variable `nombre_usuario` es **IGUAL** al string *Juan* usando un `' == '`

Efectivamente `nombre_usuario` es igual a *Juan* luego por tanto esa comparación es VERDADERA o TRUE

`nombre_usuario` no está asociado al valor *Fede*, luego el resultado es FALSO o FALSE

TEST CONDICIONALES

COMPRUEBAN SI ALGO ES CIERTO O FALSO, TRUE/FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario == 'Juan'
True
>>> nombre_usuario == 'Fede'
False
>>> █
```

Asignamos a la variable `nombre_usuario` un string con el símbolo `' = '`

Comparamos si la variable `nombre_usuario` es **IGUAL** al string *Juan* usando un `' == '`

Efectivamente `nombre_usuario` es igual a *Juan* luego por tanto esa comparación es VERDADERA o TRUE

`nombre_usuario` no está asociado al valor *Fede*, luego el resultado es FALSO o FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario != 'Juan'
False
>>> nombre_usuario != 'Fede'
True
>>> █
```


TEST CONDICIONALES

COMPRUEBAN SI ALGO ES CIERTO O FALSO, TRUE/FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario == 'Juan'
True
>>> nombre_usuario == 'Fede'
False
>>> █
```

Asignamos a la variable `nombre_usuario` un string con el símbolo `' = '`

Comparamos si la variable `nombre_usuario` es **IGUAL** al string *Juan* usando un `' == '`

Efectivamente `nombre_usuario` es igual a *Juan* luego por tanto esa comparación es VERDADERA o TRUE

`nombre_usuario` no está asociado al valor *Fede*, luego el resultado es FALSO o FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario != 'Juan'
False
>>> nombre_usuario != 'Fede'
True
>>> █
```

Comparamos si la variable `nombre_usuario` es **DIFERENTE** al string *Juan* usando un `' != '`

Como `nombre_usuario` *sí que esta asociado* a la cadena *Juan* nos da como resultado un FALSE

TEST CONDICIONALES

COMPRUEBAN SI ALGO ES CIERTO O FALSO, TRUE/FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario == 'Juan'
True
>>> nombre_usuario == 'Fede'
False
>>> 
```

Asignamos a la variable `nombre_usuario` un string con el símbolo `' = '`

Comparamos si la variable `nombre_usuario` es **IGUAL** al string *Juan* usando un `' == '`

Efectivamente `nombre_usuario` es igual a *Juan* luego por tanto esa comparación es VERDADERA o TRUE

`nombre_usuario` no está asociado al valor *Fede*, luego el resultado es FALSO o FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario != 'Juan'
False
>>> nombre_usuario != 'Fede'
True
>>> 
```

Comparamos si la variable `nombre_usuario` es **DIFERENTE** al string *Juan* usando un `' != '`

Como `nombre_usuario` *sí que esta asociado* a la cadena *Juan* nos da como resultado un FALSE

Si preguntamos si `nombre_usuario` es diferente a *Fede* nos dará un VERDADERO o TRUE

TEST CONDICIONALES

COMPRUEBAN SI ALGO ES CIERTO O FALSO, TRUE/FALSE

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario == 'Juan'
True
>>> nombre_usuario == 'Fede'
False
>>> █
```

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario != 'Juan'
False
>>> nombre_usuario != 'Fede'
True
>>> █
```

```
>>> nombre = 'Fede'
>>> 'e' in nombre
True
>>> 'a' in nombre
False
>>> 'de' in nombre
True
>>> █
```

También podemos comprobar si cierto string está contenido en otro string

TEST CONDICIONALES

STRINGS: MAYÚSCULAS Y MINÚSCULAS

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario == 'Juan'
True
>>> nombre_usuario == 'juan'
False
>>> nombre_usuario.lower() == 'juan'
True
>>> █
```

La condición es sensible a las mayúsculas y las minúsculas (*case sensitive*)

TEST CONDICIONALES

STRINGS: MAYÚSCULAS Y MINÚSCULAS

```
>>> nombre_usuario = 'Juan'
>>> nombre_usuario == 'Juan'
True
>>> nombre_usuario == 'juan'
False
>>> nombre_usuario.lower() == 'juan'
True
>>> █
```

La condición es sensible a las mayúsculas y las minúsculas (*case sensitive*)

Hacemos la comparación insensible a la forma de nombre_usuario (*case insensitive*)

TEST CONDICIONALES

COMPARACIONES NUMÉRICAS

```
>>> edad = 23
>>> edad == 23
True
>>> edad != 23
False
>>> █
```

Comprobar igualdad o desigualdad

```
>>> edad > 24
False
>>> edad < 24
True
>>> edad < 23
False
>>> edad <= 23
True
>>> █
```

TEST CONDICIONALES

COMPARACIONES NUMÉRICAS

```
>>> edad = 23
>>> edad == 23
True
>>> edad != 23
False
>>> █
```

Comprobar igualdad o desigualdad

```
>>> edad > 24
False
>>> edad < 24
True
>>> edad < 23
False
>>> edad <= 23
True
>>> █
```

Comprobar si es *mayor que* otro numero

TEST CONDICIONALES

COMPARACIONES NUMÉRICAS

```
>>> edad = 23
>>> edad == 23
True
>>> edad != 23
False
>>> █
```

Comprobar igualdad o desigualdad

```
>>> edad > 24
False
```

Comprobar si es *mayor que* otro numero

```
>>> edad < 24
True
```

Comprobar si es *menor que* otro numero

```
>>> edad < 23
False
```

```
>>> edad <= 23
True
```

```
>>> █
```


TEST CONDICIONALES

COMPARACIONES NUMÉRICAS

```
>>> edad = 23
>>> edad == 23
True
>>> edad != 23
False
>>> █
```

Comprobar igualdad o desigualdad

```
>>> edad > 24
False
```

Comprobar si es *mayor que* otro numero

```
>>> edad < 24
True
```

Comprobar si es *menor que* otro numero

```
>>> edad < 23
False
```

```
>>> edad <= 23
True
```

Comprobar si es *menor o igual que* otro numero

```
>>> █
```

TEST CONDICIONALES

COMPARACIONES NUMÉRICAS

```
>>> edad = 23
>>> edad == 23
True
>>> edad != 23
False
>>> █
```

Comprobar igualdad o desigualdad

```
>>> edad > 24
False
```

Comprobar si es *mayor que* otro numero

```
>>> edad < 24
True
```

Comprobar si es *menor que* otro numero

```
>>> edad < 23
False
```

```
>>> edad >= 23
True
```

Comprobar si es *mayor o igual que* otro numero

```
>>> █
```

TESTEAR CONDICIONES MÚLTIPLES

EI USO DE *AND* Y *OR*

Muchas veces puede ser útil comprobar si varias condiciones se cumplen simultáneamente.

En estos casos usaremos la palabra *AND*

TESTEAR CONDICIONES MÚLTIPLES

EL USO DE *AND* Y *OR*

Muchas veces puede ser útil comprobar si varias condiciones se cumplen simultáneamente.

Ejemplo: Buscamos a una persona que se llama Juan y que sabemos que es tenga 21 años o más.

En estos casos usaremos la palabra *AND*

```
>>> nombre_usuario = 'Juan'
>>> edad = 23
>>> (nombre_usuario == 'Juan') and (edad >= 21)
True
>>> █
```

TESTEAR CONDICIONES MÚLTIPLES

EL USO DE *AND* Y *OR*

Muchas veces puede ser útil comprobar si varias condiciones se cumplen simultáneamente.

Ejemplo: Buscamos a una persona que se llama Juan y que sabemos que es tenga 21 años o más.

En estos casos usaremos la palabra *AND*

```
>>> nombre_usuario = 'Juan'
>>> edad = 16
>>> (nombre_usuario == 'Juan') and (edad >= 21)
False
>>> █
```

TESTEAR CONDICIONES MÚLTIPLES

EI USO DE *AND* Y *OR*

En otras ocasiones nos basta si una de varias condiciones se cumple.

En estos casos usaremos la palabra *OR*

TESTEAR CONDICIONES MÚLTIPLES

EL USO DE *AND* Y *OR*

En otras ocasiones nos basta si una de varias condiciones se cumple.

Ejemplo: Hay un local en el que solo puede entrar gente mayor de edad o menores acompañados de un mayor de edad

En estos casos usaremos la palabra *OR*

```
>>> edad_juan = 17
>>> edad_lucas = 22
>>> (edad_juan > 18) or (edad_lucas > 18)
True
>>> edad_juan = 17
>>> edad_lucas = 16
>>> (edad_juan > 18) or (edad_lucas > 18)
False
```


TESTEAR CONDICIONES MÚLTIPLES

EL USO DE *AND* Y *OR*

En otras ocasiones nos basta si una de varias condiciones se cumple.

Ejemplo: Hay un local en el que solo puede entrar gente mayor de edad o menores acompañados de un mayor de edad

En estos casos usaremos la palabra *OR*

```
>>> edad_juan = 17
>>> edad_lucas = 22
>>> (edad_juan > 18) or (edad_lucas > 18)
True
>>> edad_juan = 17
>>> edad_lucas = 16
>>> (edad_juan > 18) or (edad_lucas > 18)
False
```

Con que uno de los dos lo cumpla será TRUE

TESTEAR CONDICIONES MÚLTIPLES

EL USO DE AND Y OR

En otras ocasiones nos basta si una de varias condiciones se cumple.

Ejemplo: Hay un local en el que solo puede entrar gente mayor de edad o menores acompañados de un mayor de edad

En estos casos usaremos la palabra OR

```
>>> edad_juan = 17
>>> edad_lucas = 22
>>> (edad_juan > 18) and (edad_lucas > 18)
False
>>> edad_juan = 17
>>> edad_lucas = 16
>>> (edad_juan > 18) or (edad_lucas > 18)
False
```

Con el AND ambos deben cumplir la condición

TESTEAR CONDICIONES MÚLTIPLES

EL USO DE *AND* Y *OR*

En otras ocasiones nos basta si una de varias condiciones se cumple.

Ejemplo: Hay un local en el que solo puede entrar gente mayor de edad o menores acompañados de un mayor de edad

En estos casos usaremos la palabra *OR*

```
>>> edad_juan = 17
>>> edad_lucas = 22
>>> (edad_juan > 18) or (edad_lucas > 18)
True
>>> edad_juan = 17
>>> edad_lucas = 16
>>> (edad_juan > 18) or (edad_lucas > 18)
False
```

Como ninguno de los dos lo cumple será FALSE

TESTEAR CONDICIONES MÚLTIPLES

EI USO DE *AND* Y *OR*

Podemos concatenar tantas condiciones como queramos:

TESTEAR CONDICIONES MÚLTIPLES

EL USO DE *AND* Y *OR*

Podemos concatenar tantas condiciones como queramos:

```
>>> edad_juan = 17
>>> edad_lucas = 22
>>> edad_jose = 28
>>> (edad_juan > 18) and (edad_lucas > 18) and (edad_jose > 18)
False
```

TESTEAR CONDICIONES MÚLTIPLES

EL USO DE *AND* Y *OR*

Podemos concatenar tantas condiciones como queramos:

```
>>> edad_juan = 17
>>> edad_lucas = 22
>>> edad_jose = 28
>>> (edad_juan > 18) and (edad_lucas > 18) and (edad_jose > 18)
False
```

Ejemplo: Hay un local en el que solo puede entrar gente mayor de edad o menores acompañados de alguien mayor de 25



TESTEAR CONDICIONES MÚLTIPLES

EL USO DE *AND* Y *OR*

Podemos concatenar tantas condiciones como queramos:

```
>>> edad_juan = 17
>>> edad_lucas = 22
>>> edad_jose = 28
>>> (edad_juan > 18) and (edad_lucas > 18) and (edad_jose > 18)
False
```

Ejemplo: Hay un local en el que solo puede entrar gente mayor de edad o menores acompañados de alguien mayor de 25

```
>>> ((edad_juan > 18) and (edad_lucas > 18) and (edad_jose > 18)) or ((edad_juan > 25) or (edad_lucas > 25) or (edad_jose > 25))
True
>>> █
```


TESTEAR CONDICIONES MÚLTIPLES

EL USO DE *AND* Y *OR*

Podemos concatenar tantas condiciones como queramos:

```
>>> edad_juan = 17
>>> edad_lucas = 22
>>> edad_jose = 28
>>> (edad_juan > 18) and (edad_lucas > 18) and (edad_jose > 18)
False
```

Ejemplo: Hay un local en el que solo puede entrar gente mayor de edad o menores acompañados de alguien mayor de 25

```
>>> ((edad_juan > 18) and (edad_lucas > 18) and (edad_jose > 18)) or ((edad_juan > 25) or (edad_lucas > 25) or (edad_jose > 25))
True
>>> █
```



LISTA DE TEST CONDICIONALES

Igualdades	— — — —	$h == 10$
Desigualdades	— — — —	$h != 10$
Mayor que	— — — —	$h > 0$
Mayor o igual que	— — — —	$h >= 0$
Menor que	— — — —	$h < 0$
Menor o igual que	— — — —	$h <= 0$

Concatenación de condicionales:

and (y) : todas las condiciones deben cumplirse

or (o): una de las condiciones debe cumplirse

EXPRESIONES BOOLEANAS

TAN SOLO ES OTRA MANERA DE LLAMAR A LOS *TEST CONDICIONALES*

Un valor booleano puede ser solo VERDADERO (TRUE) o FALSO (FALSE)

```
>>> proceso_activo = True  
>>> permiso_edicion = False
```

Estos valores pueden usarse para:

- Tener información de si un proceso se está ejecutando
- Saber si un usuario tiene ciertos permisos sobre un archivo

...

...

EXPRESIONES BOOLEANAS

TAN SOLO ES OTRA MANERA DE LLAMAR A LOS *TEST CONDICIONALES*

Un valor booleano puede ser solo VERDADERO (TRUE) o FALSO (FALSE)

```
>>> proceso_activo = True  
>>> permiso_edicion = False
```

Nos serán útiles para realizar un seguimiento del estado de nuestro programa

Estos valores pueden usarse para:

- Tener información de si un proceso se está ejecutando
- Saber si un usuario tiene ciertos permisos sobre un archivo

...

...

IF STATEMENTS = EXPRESIONES IF

Expresion Si-Entonces de PSEINT:

If Statement en Python:

IF STATEMENTS = EXPRESIONES IF

Expresion Si-Entonces de PSEINT:

```
Si expresion_logica Entonces
|   ejecucion_codigo_1
SiNo
|   ejecucion_codigo_2
FinSi
```

If Statement en Python:

IF STATEMENTS = EXPRESIONES IF

Expresion Si-Entonces de PSEINT:

```
Si expresion_logica Entonces
|   ejecucion_codigo_1
SiNo
|   ejecucion_codigo_2
FinSi
```

If Statement en Python:

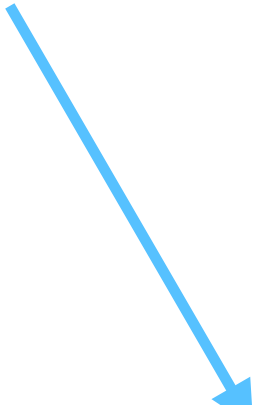
```
if conditional_1:
|   ejecucion_de_codigo_1
elif condicional_2:
|   ejecucion_de_codigo_2
else:
|   ejecucion_de_codigo_3
```


IF STATEMENTS = EXPRESIONES IF

Expresion Si-Entonces de PSEINT:

```
Si expresion_logica Entonces
    ejecucion_codigo_1
SiNo
    ejecucion_codigo_2
FinSi
```

If Statement en Python:



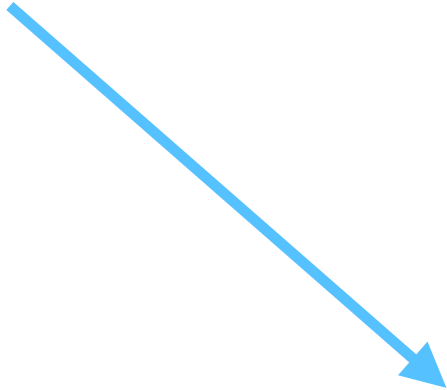
```
if conditional_1:
    ejecucion_de_codigo_1
elif condicional_2:
    ejecucion_de_codigo_2
else:
    ejecucion_de_codigo_3
```

IF STATEMENTS = EXPRESIONES IF

Expresion Si-Entonces de PSEINT:

```
Si expresion_logica Entonces
|   ejecucion_codigo_1
SiNo
|   ejecucion_codigo_2
FinSi
```

If Statement en Python:




```
if conditional_1:
|   ejecucion_de_codigo_1
elif condicional_2:
|   ejecucion_de_codigo_2
else:
|   ejecucion_de_codigo_3
```

IF STATEMENTS = EXPRESIONES IF

Expresion Si-Entonces de PSEINT:

```
Si expresion_logica Entonces
|   ejecucion_codigo_1
SiNo
|   ejecucion_codigo_2
FinSi
```

If Statement en Python:




```
if conditional_1:
|   ejecucion_de_codigo_1
elif condicional_2:
|   ejecucion_de_codigo_2
else:
|   ejecucion_de_codigo_3
```

IF STATEMENTS = EXPRESIONES IF

Expresion Si-Entonces de PSEINT:

```
Si expresion_logica Entonces
    ejecucion_codigo_1
SiNo
    ejecucion_codigo_2
FinSi
```

If Statement en Python:




```
if conditional_1:
    ejecucion_de_codigo_1
elif conditional_2:
    ejecucion_de_codigo_2
else:
    ejecucion_de_codigo_3
```

IF STATEMENTS = EXPRESIONES IF

Expresion Si-Entonces de PSEINT:

```
Si expresion_logica Entonces
    ejecucion_codigo_1
SiNo
    ejecucion_codigo_2
FinSi
```

If Statement en Python:



```
if conditional_1:
    ejecucion_de_codigo_1
elif condicional_2:
    ejecucion_de_codigo_2
else:
    ejecucion_de_codigo_3
```


IF STATEMENTS = EXPRESIONES IF

Expresion Si-Entonces de PSEINT:

```
Si expresion_logica Entonces
    ejecucion_codigo_1
SiNo
    ejecucion_codigo_2
FinSi
```

If Statement en Python:

```
if conditional_1:
    ejecucion_de_codigo_1
elif condicional_2:
    ejecucion_de_codigo_2
else:
    ejecucion_de_codigo_3
```

IF STATEMENTS = EXPRESIONES IF

Expresion Si-Entonces de PSEINT:

```
Si expresion_logica Entonces
    ejecucion_codigo_1
SiNo
    ejecucion_codigo_2
FinSi
```

If Statement en Python:

```
if conditional_1:
    ejecucion_de_codigo_1
elif condicional_2:
    ejecucion_de_codigo_2
else:
    ejecucion_de_codigo_3
```

La **INDENTACION** es necesaria en Python

IF STATEMENTS = EXPRESIONES IF

PSEINT

Python

IF STATEMENTS = EXPRESIONES IF

PSEINT

```
//Definición e inicialización
//de variable precio
Definir precio Como Entero
precio = 0
//Leemos el nuevo dato y se
//almacena en "precio"
Leer precio
//Condición SI-ENTONCES
Si precio < 20 Entonces
    Escribir "El precio es menor a 20. Es: ", precio
SiNo
    Escribir "El precio es mayor o igual a 20. Es: ", precio
FinSi
```

Python

IF STATEMENTS = EXPRESIONES IF

PSEINT

```
//Definición e inicialización
//de variable precio
Definir precio Como Entero
precio = 0
//Leemos el nuevo dato y se
//almacena en "precio"
Leer precio
//Condición SI-ENTONCES
Si precio < 20 Entonces
    Escribir "El precio es menor a 20. Es: ", precio
SiNo
    Escribir "El precio es mayor o igual a 20. Es: ", precio
FinSi
```

Python

```
#Leemos el nuevo dato
#almacena en precio
precio = int(input())
# condicional IF STATEMENT
if precio < 20:
    print('El precio es menor a 20. Es ', precio)
else:
    print('El precio es mayor a 20. Es ', precio)
```



IF STATEMENTS ANIDADOS

PSEINT

Python

IF STATEMENTS ANIDADOS

PSEINT

Python

```
Si expresion_logica_1 Entonces
    Si expresion_logica_2 Entonces
        ejecucion_codigo_1
    SiNo
        ejecucion_codigo_2
    FinSi
SiNo
    Si expresion_logica_3 Entonces
        ejecucion_codigo_3
    SiNo
        ejecucion_codigo_4
    FinSi
FinSi
```

IF STATEMENTS ANIDADOS

PSEINT

```
Si expresion_logica_1 Entonces
    Si expresion_logica_2 Entonces
        ejecucion_codigo_1
    SiNo
        ejecucion_codigo_2
    FinSi
SiNo
    Si expresion_logica_3 Entonces
        ejecucion_codigo_3
    SiNo
        ejecucion_codigo_4
    FinSi
FinSi
```

Python

```
if condicional_1:
    if condicional_a:
        ejecucion_de_codigo_a
    else:
        ejecucion_de_codigo_b
else:
    if condicional_c:
        ejecucion_de_codigo_c
    elif condicional_d:
        ejecucion_de_codigo_d
    else:
        ejecucion_de_codigo_e
```



IF STATEMENTS ANIDADOS

PSEINT

Python

IF STATEMENTS ANIDADOS

PSEINT

```
//Definición e inicialización
Definir hasPagado Como Logica
Definir precio Como Entero
hasPagado = Falso
precio = 0
//Leemos los datos
Leer hasPagado
Leer precio
//Condición SI-ENTONCES anidado
Si hasPagado = Falso Entonces
    Escribir "No has pagado aún"
    Si precio ≤ 20 Entonces
        Escribir "Tienes que pagar menos de 20 euros"
    SiNo
        Escribir "Tienes que pagar más de 20 euros"
    FinSi
SiNo
    Escribir "Ya has pagado"
FinSi
```

Python

IF STATEMENTS ANIDADOS

PSEINT

```
//Definición e inicialización
Definir hasPagado Como Logica
Definir precio Como Entero
hasPagado = Falso
precio = 0
//Leemos los datos
Leer hasPagado
Leer precio
//Condición SI-ENTONCES anidado
Si hasPagado = Falso Entonces
    Escribir "No has pagado aún"
    Si precio ≤ 20 Entonces
        Escribir "Tienes que pagar menos de 20 euros"
    SiNo
        Escribir "Tienes que pagar más de 20 euros"
    FinSi
SiNo
    Escribir "Ya has pagado"
FinSi
```

Python

```
# Leemos los datos
# que necesitamos de entrada
hasPagado = input()
precio = float(input())
#condicional IF anidado
if hasPagado == "False":
    print("No has pagado aún")
    if precio <= 20:
        print("Tienes que pagar menos de 20 euros")
    else:
        print("Tienes que pagar más de 20 euros")
else:
    print("Ya has pagado")
```

ESTRUCTURA SEGÚN O SWITCH-CASE

PSEINT

Python

ESTRUCTURA SEGÚN O SWITCH-CASE

PSEINT

Python

```
Definir nombre Como Texto
nombre = ""

Escribir "Introduce tu nombre"
Leer nombre

Segun nombre Hacer
    "Juan":
        Escribir "Bienvenido Juan!"
    "Maria":
        Escribir "Bienvenida Maria!"
    "Pepa":
        Escribir "Bienvenida Pepa!"
De otro Modo:
    Escribir "Bienvenido, seas quien seas!"
FinSegun
```

ESTRUCTURA SEGÚN O SWITCH-CASE

PSEINT

```
Definir nombre Como Texto
nombre = ""

Escribir "Introduce tu nombre"
Leer nombre

Segun nombre Hacer
    "Juan":
        Escribir "Bienvenido Juan!"
    "Maria":
        Escribir "Bienvenida Maria!"
    "Pepa":
        Escribir "Bienvenida Pepa!"
De otro Modo:
    Escribir "Bienvenido, seas quien seas!"
FinSegun
```

Python

En Python no existe la estructura switch-case (según)



Podemos imitar su funcionamiento con el if-statement

ESTRUCTURA SEGÚN O SWITCH-CASE

PSEINT

```
Definir nombre Como Texto
nombre = ""

Escribir "Introduce tu nombre"
Leer nombre

Segun nombre Hacer
    "Juan":
        Escribir "Bienvenido Juan!"
    "Maria":
        Escribir "Bienvenida Maria!"
    "Pepa":
        Escribir "Bienvenida Pepa!"
De otro Modo:
    Escribir "Bienvenido, seas quien seas!"
FinSegun
```

Python

```
# leemos el nombre
nombre = input('Introduce tu nombre')

# usamos condicional IF
if nombre == 'Juan':
    print('¡Bienvenido, Juan!')
elif nombre == 'Maria':
    print('¡Bienvenida, Maria!')
elif nombre == 'Pepa':
    print('¡Bienvenida, Pepa!')
else:
    print('¡Bienvenido, seas quien seas!')
```



REPASO

1. Test condicionales simples
2. Test condicionales múltiples
3. Expresión If simple
4. Expresión If anidada
5. Sustitución del switch case por expresiones If

CÔNQUER BLOCKS