

UNIVERSIDAD DE ALCALÁ

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN



**LABORATORIO
FUNDAMENTOS DE LA PROGRAMACIÓN
2ª PARTE TEMA 1**

2. SENTENCIAS CONDICIONALES

EXPRESIONES CONDICIONALES.

El flujo de ejecución de un programa en Python es estrictamente secuencial, no obstante, se puede alterar para hacer que tomen decisiones a partir de datos y/o resultados intermedios y en función del resultado se ejecuten ciertas sentencias y otras no. Este tipo de alteración del flujo de control se consigue con **sentencias condicionales** o de selección.

Antes de explicar las diferentes estructuras para cambiar el flujo de ejecución de un programa tenemos que recordar los operadores lógicos resumidos en la siguiente tabla:

donde V = TRUE F= FALSE

| = Negación. $\wedge$ = y lógico $\vee$ = ó lógico

1		2	3	3	4
p	q	$\neg q$	$p \vee q$	$\neg q \vee p$	$((p \vee q) \wedge (\neg q \vee p))$
V	V	F	V	V	V
F	V	F	V	F	F
V	F	V	V	V	V
F	F	V	F	V	F

En esta tabla podemos observar 4 ejemplos de operaciones booleanas:

- 1) Sean p y q variables booleanas inicializadas con valores True o False.
- 2) Negación de variable q : not q.
- 3) Operador or p y q, posteriormente (not q or p).
- 4) Operador and de los dos resultados de las operaciones anteriores.

2.1. INSTRUCCIÓN CONDICIONAL IF ... ELSE.

➤ **Sintaxis:**

```
if condición:

    instrucciones si se cumple la condición

else:

    instrucciones si no se cumple la condición
```

La primera línea contiene la expresión lógica a evaluar, y siempre se termina la línea con dos puntos (:). Es muy importante el tabulado, ya que como explicamos en el tema anterior la forma de diferenciar los bloques es la **indentación** que precisamente es tabular con 4 espacios lo que pertenece a un mismo bloque. La cláusula **else** puede ser opcional.

1- Programa con instrucción condicional omitiendo cláusula else:

Realizar un programa que escriba mensaje “Eres universitario” si la persona que lo ejecuta tiene más de 18 años de edad.

```
# Archivo: ejemplo12.py
# Autor: XXXXX XXXXXXXX (nombre y apellidos)
# Fecha: DD/MM/AAAA
# Descripción: Realizar un programa que escriba 'Eres
# universitario' si la persona tiene más de 18 años.

# Programa principal
edad = int(input('Introduce tu edad: '))

if edad >= 18:
    print('Eres universitario') #sangría de 4 espacios
```

2- Programa con instrucción condicional con cláusula else.

Realizar un programa que dados 3 números los ordene de mayor a menor.

```
# Archivo: ejemplo13.py
# Autor: XXXXX XXXXXXXX (nombre y apellidos)
# Fecha: DD/MM/AAAA
# Descripción: Realizar un programa que dados 3 números los
# ordene de mayor a menor.

# Programa principal
x = int(input('Dame el primer número entero: '))
y = int(input('Dame el segundo número entero: '))
z = int(input('Dame el tercero número entero: '))
if x >= y:
    print('')# esta sentencia no hace nada
else:
```

```
    r = x
    x = y
    y = r
if x >= z:
    print('')
else:
    r = x
    x = z
    z = r
if y >= z:
    print('')
else:
    r = y
    y = z
    z = r
print( x,y,z)
```

3- Programa con instrucciones condicionales anidadas.

Realizar un programa que dados tres números obtenga cuál de ellos es menor.

```
# Archivo: ejemplo14.py
# Autor: XXXXX XXXXXX (nombre y apellidos)
# Fecha: DD/MM/AAAA
# Descripción: Realizar un programa que dados 3 números
# diga cuál de ellos es menor.

# Programa principal
n1 = int(input('Dame el primer numero entero: '))
n2 = int(input('Dame el segundo numero entero: '))
n3 = int(input('Dame el tercero numero entero: '))
if n1 < n2:
    if n1 < n3:
        menor = n1
    else:
        menor = n3
else:
    if n2 < n3:
        menor = n2
    else:
        menor = n3
print('El número menor es ', menor)
```

2.2. OTRO TIPO DE INSTRUCCIONES CONDICIONALES.

Cuando hay más de dos posibles alternativas, permite encadenar varias condiciones y así poder abarcar un mayor número de resultados, se utilizará la siguiente instrucción.

➤ **Sintaxis de if... elif... else...:**

Este tipo de sentencias permiten encadenar varias condiciones. La palabra clave **elif** es una contracción de **else if**. Es similar a la instrucción **case** de C++, se pueden valorar muchas condiciones.

```
if condición:

    instrucciones si se cumple la condición

elif condicion:

    instrucciones si se cumple la nueva condición

else:

    instrucciones si no se cumple ninguna condición
```

Ejemplo 4 :

Realizar un programa que dado un número del 1 al 5 obtenga su valor.

```
# Archivo: ejemplo14b.py
# Autor: XXXXX XXXXXXXX (nombre y apellidos)
# Fecha: DD/MM/AAAA
# Descripción: Realizar un programa que dado un número del
#1 al 5 saca su valor por pantalla.

# Programa principal
n = int(input(' Dame  un número del uno al 5: '))
if n == 1:
    print('El número leído es', n)
elif n == 2:
    print('El número leído es', n)
elif n == 3:
    print('El número leído es', n)
elif n == 4:
    print('El número leído es', n)
elif n == 5:
    print('El número leído es', n)
else:
    print('Error al introducir número')
```

➤ Sintaxis de if... else anidados:

Es muy importante la **indentación**. La forma de declarar en Python los bloques de programas o funciones es con un sangrado de 4 espacios.

En la sintaxis se observa que la segunda línea tiene un sangrado de 4 espacios respecto a la anterior, la tercera línea que pertenece a ese mismo bloque tiene el mismo sangrado, en cambio la siguiente línea tiene un sangrado de 4 espacios respecto a la anterior ya que se trata de un if dentro de otro if. Entre el primer if, y el tercer if hay una diferencia de 8 espacios (se ha indentado 2 veces, una por cada if).

```
if condición:
    instrucciones si se cumple la condición
    if condición:
        instrucción
        if condición:
            instrucción
        else:
            instrucción
    else:
        instrucción
else:
    instrucciones si no se cumple la condición
```



EJERCICIOS LABORATORIO

FUNDAMENTOS DE LA PROGRAMACIÓN

TEMA1 Parte 2:

INSTRUCCIONES CONDICIONALES

2ª PARTE.TEMA 1: Instrucciones Condicionales

ENUNCIADOS DE EJERCÍCIOS INSTRUCCIONES CONDICIONALES.

1. Escribir un programa que lea la calificación numérica de un ejercicio (de 0 a 10) y la devuelva en notación alfabética (suspense, aprobado, notable, sobresaliente).
2. Escribir un programa que lea dos caracteres y averigüe si se introdujeron en el orden de su código ASCII o no. Se deberá además comprobar si el primero de ellos es una cifra, y en caso afirmativo indicar si es impar y si es o no primo.
3. Escribir un programa que pida una fecha de nacimiento (día, mes, año) y calcule la edad correspondiente. (*) necesito la fecha actual, mirar tabla de ayuda, para obtenerla se ha de invocar a librería datetime.
4. Escribir un programa que determine si un año es o no bisiesto. Son bisiestos todos los años que son múltiplos de 4 excepto aquellos que además son múltiplos de 100 y no lo son de 400. (ej: 1996 es bisiesto, 2000 es bisiesto pero 1800 no lo es).
5. Reutilizar el código del programa anterior para escribir un programa que pida un mes y un año y devuelva el número de días de ese mes.
6. Dadas las coordenadas de un punto en el plano, ubicarlo en él (4 ejes y 4 cuadrantes).
7. Dados tres puntos del plano averiguar cuales están más cercanos.
8. Escribir un programa que lea la hora en notación de 24 horas y la devuelva en notación de 12 horas (ejemplo: las 18:30 serán las 6:30 PM).
9. Una compañía de alquiler de automóviles desea un programa para emitir sus facturas, con las siguientes consideraciones:
 1. Cantidad fija de 120 euros. si no se rebasan los 300 Km.
 2. Si la distancia recorrida es mayor que 300 Km:
 - si la distancia es menor o igual que 1000 Km, se cobrarán las 120euros más el kilometraje que exceda de 300 Km a razón de 5euros/Km.
 - si la distancia es mayor que 1000 Km, se cobrarán las 120euros más el kilometraje a razón de 5euros/Km para los kilómetros entre el 300 y el 1000 y 3euros/Km para el resto.
10. Obtener las raíces de una ecuación de segundo grado $ax^2 + bx + c = 0$ a partir de sus coeficientes a , b y c .

11. Escribir un programa que acepte un año escrito en cifras arábigas y lo visualice escrito en números romanos, dentro del rango 1 a 2000. (I=1, V=5, X=10, L=50, C=100, D=500, M=1000).
12. Dado un punto del plano, averiguar el ángulo que forma la parte positiva del eje de las abscisas con el vector que une el origen de coordenadas con el punto dado.
13. Escribir un programa por el que se introducen dos números por teclado y una operación y devuelva el resultado de la misma.

Funciones predefinidas necesarias para afrontar los ejercicios 2ª Parte Tema1

n1 % n2→Dados 2 variables numéricas n1 y n2 el Operador %, es el resto que resulta de dividir el contenido de la variable n1 entre la n2.

chr(n)→Siendo n un número entero entre 0 y 255, devuelve el carácter ASCII correspondiente al número x.

ord(c)→Sea c de tipo **char**, ord(c) devuelve el valor ASCII correspondiente al valor de la variable c.

sqrt(x)→Sea x un número real sqrt(x) devuelve la Raíz cuadrada ($\sqrt{x}$) (por tanto otro real). Esta función pertenece a la librería math, por ese motivo tenéis que invocarla antes de poder utilizarla

Instrucciones para obtener la fecha y hora del sistema

1º- Invocamos la librería data time con la siguiente sentencia

from datetime import * (Al invocar esta librería podremos acceder a todas las funciones necesarias para saber la hora y fecha del sistema y poder trabajar con ella)

2º- Llamamos a la función datetime implementada en dicha librería

a = datetime.now() --> Obtiene fecha y hora actual del sistema. Proporciona el año, mes, día, hora, minutos, segundos y microsegundos.

Ejemplo de uso:

```
from datetime import *
ahora = datetime.now()
print ("Fecha y hora actuales:")
print ("Año actual: %d" % ahora.year)
print ("Mes actual: %d" % ahora.month)
print ("Día actual: %d" % ahora.day)
print ("Hora actual: %d" % ahora.hour)
print ("Minuto actual: %d" % ahora.minute)
print ("Segundo actual: %d" % ahora.second)
print ("Microsegundo actual: %d" % ahora.microsecond)
```