

ESTRUCTURA DE CONTROL CONDICIONAL: IF-ELSE

Si las variables le dan “memoria” al programa, los condicionales le otorgan “inteligencia”. Gracias a los condicionales, el computador puede “tomar una decisión” respecto de si ejecutar o no un conjunto de instrucciones. La forma más simple de un condicional es así:

Si se cumple una condición lógica, ejecute el conjunto 1 de instrucciones
Sino, ejecuta el conjunto 2 de instrucciones

Por ejemplo:

```
Si (el pronóstico para hoy es soleado)(condición)
    me alegro(instrucción1 del Si)
    me visto con ropa ligera(instrucción2 del Si)
Si no(Nota que aquí no hay condición explícita)
    me enoja(instrucción 1 del Sino)
    me visto con ropa abrigada(instrucción2 del Sino)
```

Nota que al decir “Si no”, no necesito establecer una condición para que se ejecute el conjunto 2 de instrucciones. Por lógica, “Si no”, significa “Si no se cumple la condición anteriormente establecida”.

Existen diferentes tipos de estructuras de control, pero todas ellas tienen en común el uso de condiciones lógicas para determinar si se ejecutan o no las instrucciones asociadas a ellas.

Normalmente, las condiciones lógicas permiten evaluar una (o más) preguntas sobre el valor de las variables. Por ejemplo, la condición lógica ($tmp > 32$) es VERDADERA si la variable `tmp` tiene almacenado un valor mayor a 32 y FALSA si tiene almacenado un valor menor o igual a 32. También podemos emplear condiciones compuestas usando de los operadores lógicos `and`, `or`. Por ejemplo, la condición lógica ($tmp > 32$ `and` $nivel_agua < 20$) es VERDADERA si la variable `tmp` tiene almacenado un valor mayor a 32 y si la variable `nivel_agua` es menor a 20. De lo contrario, la condición lógica es FALSA.

La sintaxis de la estructura condicional estándar en un condicional en Python es la siguiente:

```
if ( <condición(es) una o más> ) :
    // instrucciones del bloque if
else:
    // instrucciones del bloque else
```

La estructura general se compone de la siguiente forma¹: El `if` tiene asociada una condición lógica, que puede ser VERDADERA o FALSA. Si la condición es VERDADERA, entonces el control del programa ejecutará el bloque de instrucciones `if`. En cambio, si la condición es FALSA, el control del programa ejecutará el bloque de instrucciones `else`.

Nota que no existe una condición explícita para el `else`. Solo el `if` tiene asociada una condición lógica explícita. La condición del `else` es implícita (es la negación de la condición del `if`). Es decir, el bloque de instrucciones `else` se ejecuta si no se cumple la condición explícita asociada al `if`. Este

¹ Supón que este código está contenido en un programa.

es un error común de los principiantes: print(una condición para **else**. La parte del **else** no necesita condición explícita.

Nota también que cuando el control del programa escoge un camino, no puede regresar a ejecutar las instrucciones del bloque que no eligió. Por ello el programador debe definir correctamente el curso de acción que seguirá su programa cuando se encuentre en ejecución.

Aparte de la estructura estándar **if...else...**, existen varias maneras de usar la estructura condicional. Las describiremos a continuación.

if...

Aunque una manera de referirse a los condicionales es llamándolos **if-else**, no es obligación usar un **else** por cada **if** del programa. Si solo es necesario actuar (ejecutar instrucciones) en caso que se cumpla una condición (y hacer nada en caso que no se cumpla), se usa solo **if**.

Ejemplo: se solicita al usuario que ingrese nombre, apellido y edad. Si tiene menos de 18 años, se despliega un mensaje determinado. Sino, no hay acciones que ejecutar.

```
print( "Ingresa por favor tu nombre, apellido y edad" )
nombre = input()
apellido = input()
edad = int( input() )
if ( edad<18 ):
    print( nombre, " ,recuerda pedir el código de autorización de tu tutor"
```

Nota que si la persona tiene 18 años o más , no hay que desplegar ningún mensaje. Por lo tanto, en este caso el **else** no es necesario.

if...else... (Sin anidación)

Cuando hay dos bloques de acciones distintas que deben realizarse dependiendo de si se cumple o no la condición lógica asociada al **if**, entonces se usa la estructura **if...else...**. En esta sección veremos la estructura simple (un **if** seguido de un **else**). Esto ocurre cuando solo se debe tomar una sola decisión.

Ejemplo: se solicita al usuario que ingrese un número y se despliega por pantalla si acaso el número es o no múltiplo de 3.

```
print( "Ingresa un número por favor" )
num = int( input() )
if ( num%3 == 0 ):
    print( "El número ingresado es múltiplo de 3" )
else:
    print( "El número ingresado no es múltiplo de 3" )
```

if...else... ANIDADOS

Cuando es necesario volver a tomar una decisión que depende del resultado de una decisión anterior, se usan estructuras condicionales anidadas (una dentro de otra).

Ejemplo: En un restaurant hay 2 menús: El menú 1 es cazuela y el Menú 2 es pescado frito (elegir entre el Menú 1 y el Menú 2 corresponde a la primera decisión). Si eliges el Menú 1, debes luego

tomar una **segunda decisión**: elegir si quieres cazuela de vacuno o de ave (esta segunda decisión solo aparece si la primera decisión fue Menú 1). En cambio, si eliges el Menú 2, debes luego decidir si quieres congrio o reineta.

Este tipo de caso se puede visualizar como un árbol de decisión, como se muestra en la Figura 1.

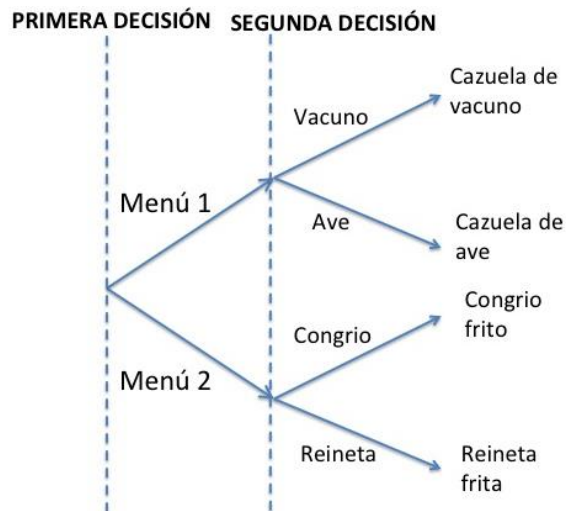


Figura 1. Ejemplo de un árbol de decisión

Un posible código para el problema anterior sería el siguiente:

```

print( "Elige tu menú: 1. Cazuela  2. Pescado frito" )
opcion = int( input() )
if ( opcion == 1 ):
    print( "Ingresa 1 si quieres cazuela de vacuno y 2 si quieres cazuela de ave" )
    cazuela = int( input() )
    if ( cazuela == 1 ):
        print( "Preparando cazuela de vacuno" )
    else:
        print( "Preparando cazuela de ave" )
else:
    print( "Ingresa 1 si quieres congrio y 2 si quieres reineta" )
    pescado = int( input() )
    if ( pescado == 1 ):
        print( "Preparando congrio frito" )
    else:
        print( "Preparando reineta frita" )
  
```

Se usa el adjetivo "anidado" porque hay una estructura `if...else...` dentro (anidada) de otra. No es necesario que haya exactamente una estructura `if...else...` dentro del `if` y del `else`. Se pueden usar combinaciones distintas. Por ejemplo:

```
if ( <condición_1> )  
    if ( <condición_2> ) :  
        //entra solo si condición 1 Y condición 2 son verdaderas  
    else:  
        //entra solo si condición 1 es verdadera Y 2 es falsa  
else:  
    // entra si la condición 1 es falsa. Nunca evalúa la condición 2
```

OJO: no todas las decisiones múltiples corresponden a `if...else...` anidados. Si una segunda decisión no depende del resultado de la primera, no se trata de estructuras anidadas, else secuenciales (una después de la otra) como veremos a continuación.

VARIOS `if` (SIN `else`)

Esta estructura se usa generalmente para construir un menú donde el usuario debe elegir una de las opciones. Si son pocas opciones (2-4), se suele usar varios `if`.

Ejemplo: pedirle al usuario que seleccione su rango de edad y desplegar un mensaje distinto dependiendo de la elección del usuario.

```
print( "Seleccione su rango de edad" )  
print("1. Menor de 18 años" )  
print("2. Entre 18 y 64 años, ambos inclusive" )  
print("3. Mayor de 65 años" )  
respuesta = int( input() )  
if (respuesta == 1):  
    print("Lo siento, eres menor de edad. Fuera de aquí" )  
  
if (respuesta == 2):  
    print("Bienvenido a Santa Agustina" )  
  
if (respuesta == 3):  
    print("Volumen no apto para sus oídos" )
```

Nota: Como esta estructura obliga al computador a evaluar todas las opciones, aún cuando el usuario haya elegido la primera, algunos prefieren de todos modos usar una estructura `if...else...` en estos casos:

```
print( "Seleccione su rango de edad" )  
print("1. Menor de 18 años" )  
print("2. Entre 18 y 64 años, ambos inclusive" )
```

```
print("3. Mayor de 65 años" )  
respuesta = int( input() )  
if (respuesta == 1):  
    print("Lo siento, eres menor de edad. Fuera de aquí" )  
else:  
    if (respuesta == 2):  
        print("Bienvenido a Santa Agustina" )  
    else:  
        print("Volumen no apto para sus oídos" )
```

Ya que así, si el usuario elige la respuesta 1, el resto del código no se ejecuta. En este curso, cualquiera de las 2 opciones se considera correcta (aun cuando la segunda es más rápida de ejecutar, pero en un programa tan pequeño como este la diferencia es imperceptible).

PROBLEMAS PROPUESTOS

1. Para el siguiente código en Python:

```
#Algoritmo Acceso
print("Ingresa tu edad:")
edad = int(input())
if (edad<18):
    print("Acceso denegado")
else:
    if (edad<25) :
        print("Pase por caja 1")
    else:
        if (edad<30) :
            print("Pase por caja 2")
        else:
            print("Pase por caja 3")
```

- a) Escribe lo que se despliega en pantalla al ejecutar código cuando el usuario ingresa el valor 25.
- b) ¿Qué conjunto de valores hace que el programa despliegue el mensaje "Pase por caja 3"?

Solución:

- a) Pase por caja 2
- b) El conjunto de los valores mayores o iguales a 30.

2. Escribe el código de un algoritmo que solicita las notas de dos pruebas, calcula el promedio y luego despliega 3 mensajes distintos dependiendo del promedio: "Felicitaciones, vas camino a aprobar" (si el promedio es mayor o igual a 4.0); "Atención, vas camino a reprobar" (promedio mayor o igual a 3.0 pero menor a 4.0) y "Pocas posibilidades de aprobar" (para promedio menor a 3.0).

Una solución posible para este programa es:

```
print( "la nota 1" )
nota1 = float( input() )
print( "la nota 2" )
nota2 = float( input() )
promedio = (nota1+nota2)/2
print ( "tu promedio es ", promedio )
if ( promedio >=4.0 ):
    print ( "Felicitaciones, vas camino a aprobar" )
else:
    if ( promedio>=3.0 ):
        print( "Atención: camino a reprobar" )
    else:
        print( "Pocas posibilidades de aprobar" )
```

3. Escribe el código de un algoritmo que solicite por pantalla el número entero A, y luego el número entero B. Luego, debe verificar si A/B produce una división entera (es decir, el resto de la división es 0). Si este es el caso, deberá mostrar por pantalla "B divide exactamente a A". En caso contrario "B no divide a A en forma".

Una solución posible para este problema es:

```
print( "Ingresa A: " )  
A = int( input() )  
print( "Ingresa B: " )  
B = int( input() )  
division = A/B  
enterodiv = int( division )  
if ( A - enterodiv*B == 0 ):  
    print( B, " divide exactamente ", A )  
else:  
    print( B, " no divide a ", A, " en forma entera" )
```


4. Escribe el código de un algoritmo que solicita los valores a,b y c de una ecuación cuadrática de la forma:

$$y = ax^2 + bx + c$$

y luego despliegue por pantalla los siguientes mensajes:

- "No hay soluciones reales" (si el discriminante es negativo)
- "Hay una solución real y es igual a " y desplegar el valor (cuando el discriminante es igual a 0)
- "Hay dos soluciones reales iguales a " y desplegar sus valores (cuando el discriminante es positivo).

Recuerda que las soluciones se calculan con la siguiente fórmula.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

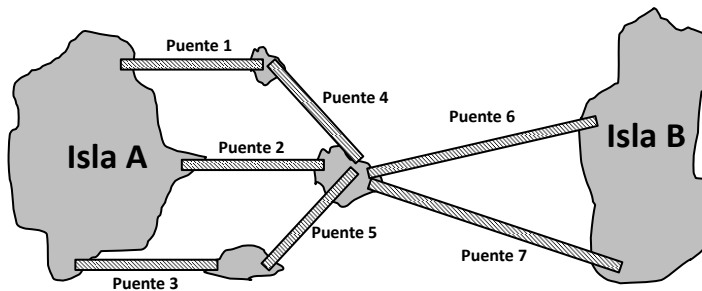
Una solución posible para este programa es

```
print( 'Ingrese los valores a, b y c de una ecuación cuadrática' )
print( 'ingrese a' )
a = int( input() )
print( 'ingrese b' )
b = int( input() )
print( 'ingrese c' )
c = int( input() )

disc = b**2-4*a*c

if disc < 0 :
    print( 'no hay soluciones reales' )
else:
    if disc == 0:
        x1 = (-b + (disc**0.5))/(2*a)
        print( 'hay una solución real y esta es: x1=', x1 )
    else:
        x1 = (-b + (disc**0.5))/(2*a)
        x2 = (-b - (disc**0.5))/(2*a)
        print( 'hay dos soluciones reales estas son: x1=', x1, ' x2=', x2 )
```

5. La Figura 1 muestra un sistema de puentes que conecta las islas A y B.



Los puentes son de tipo mecánico y el clima en esta área geográfica es muy hostil. Por lo tanto, es común que los puentes no estén operativos. Escribe un programa en Python que solicite el estado de cada uno de los puentes (0: no operativo, 1: operativo) e informe si es posible viajar desde la isla A hasta la isla B (es decir, si existe al menos un camino de puentes en buen estado que conecte ambas islas).

252

Una solución posible

La clave para resolver este problema está en determinar las condiciones lógicas que definen si al menos una ruta está operativa. Hay varias maneras de establecer estas condiciones. Una manera es la siguiente:

- No es posible viajar de A a B si:
 - o Los puentes 1, 2 y 3 están cortados (no importa el estado del resto de los puentes)
 - o Los puentes 2, 4 y 5 están cortados (no importa el estado del resto de los puentes)
 - o Los puentes 6 y 7 están cortados (no importa el estado del resto de los puentes)

Es decir:

(puentes 1, 2 y 3 malos) O (puentes 2, 4 y 5 malos) O (puentes 6 y 7 malos).

Pasos a seguir (algoritmo):

Paso 1: Solicitar estado de los puentes

Paso 2:

Si (puentes 1, 2 y 3 malos) O (puentes 2, 4 y 5 malos) O (puentes 6 y 7 malos), desplegar "No es posible cruzar de la isla A a la B".

else, desplegar "Es posible cruzar de A a B"

```
print( "Ingresa el estado de los puentes del 1 al 7" )
p1 = int( input( "p1" ) )
p2 = int( input( "p2" ) )
p3 = int( input( "p3" ) )
p4 = int( input( "p4" ) )
p5 = int( input( "p5" ) )
p6 = int( input( "p6" ) )
p7 = int( input( "p7" ) )
if ( ( p1=0 and p2=0 and p3=0 ) or ( p2=0 and p4=0 and p5=0 ) or ( p6=0 and p7=0 ) ):
    print( "No es posible cruzar entre las islas" )
else:
    print( "Es posible cruzar entre las islas" )
```

6. Escribe un programa en Python que determine si una fecha (se ingresa día, mes y año) ocurre antes o después que otra.

Una solución posible

Pasos a seguir (algoritmo):

1. Solicitar datos al usuario y almacenarlos en variables
2. Si año1 es menor que año 2, desplegar "fecha1 antes que fecha 2"
else:
 - a. Si año2 es menor que año 1, desplegar "fecha2 antes que fecha1"
else (los años son iguales, hay que discriminar por mes):
 - i. Si mes1 menor que mes2, desplegar "fecha 1 antes que fecha 2"
else:

Si mes2 menor que mes 1, desplegar "fecha2 antes que fecha 1"

else (meses iguales, discriminar por día):

Si día 1 menor que día 2, "fecha1 menor que fecha 2"

else:

Si día 1 mayor que día 2, "fecha2 antes que fecha 1"

else, "Mismas fechas"

```
print("Ingresa dia1, mes1 y año1")
dia1 = int( input( "dia" ) )
mes1 = int( input( "mes" ) )
año1 = int( input( "año" ) )

print("Ingresa dia2, mes2 y año2")
dia2 = int( input( "dia" ) )
mes2 = int( input( "mes" ) )
año2 = int( input( "año" ) )

if ( año1 < año2 ):
    print( "Fecha 1 antes que fecha 2" )
else:
    if ( año2 < año1 ):
        print( "Fecha 2 antes que fecha 1" )
    else: #años iguales
        if ( mes1 < mes2 ):
            print( "Fecha 1 antes que fecha 2" )
        else:
            if ( mes2 < mes1 ):
                print( "Fecha 1 antes que fecha 2" )
            else: #años y meses iguales
                if ( dia1 < dia2 ):
                    print( "Fecha 1 antes que fecha 2" )
                else:
                    if ( dia2 < dia1 ):
                        print( "Fecha 2 antes que fecha 1" )
                    else:
                        print( "Fechas iguales" )
```