**Condicional if**

La sentencia **if** («si») ejecuta las instrucciones sólo si se cumple una condición. Si la condición es falsa, no se hace nada:

La sintaxis es la siguiente:

if condición:

sentencias

Por ejemplo, el siguente programa felicita a alguien que aprobó la asignatura:

nota = int(raw\_input('Ingrese su nota: '))

**if** nota >= 55:

**print** 'Felicitaciones'

Ejecute este programa, probando varias veces con valores diferentes.

**Condicional if-else**

La sentencia **if-else** («si-o-si-no») decide qué instrucciones ejecutar dependiendo si una condición es verdadera o falsa:

La sintaxis es la siguiente:

if condición:

qué hacer cuando la condición es verdadera

else

qué hacer cuando la condición es falsa

Por ejemplo, el siguiente programa indica a alguien si es mayor de edad:

edad = int(raw\_input('Cual es su edad? '))

**if** edad < 18:

**print** 'Usted es menor de edad'

**else**:

**print** 'Usted es adulto'

El siguiente programa realiza acciones distintas dependiendo de si el número de entrada es par o impar:

n = int(raw\_input('Ingrese un numero: '))

**if** n % 2 == 0:

**print** 'El numero es par'

**print** 'La mitad del numero es', n / 2

**else**:

**print** 'El numero es impar'

**print** 'El sucesor del numero es', n + 1

**print** 'Listo'

La última sentencia no está indentada, por lo que no es parte del condicional, y será ejecutada siempre.

**Ciclo while**

El ciclo **while** («mientras») ejecuta una secuencia de instrucciones mientras una condición sea verdadera:

Cada una de las veces que el cuerpo del ciclo es ejecutado se llama **iteración**.

La condición es evaluada antes de cada iteración. Si la condición es inicialmente falsa, el ciclo no se ejecutará ninguna vez.

La sintaxis es la siguiente:

while condición:

sentencias

Por ejemplo, el siguiente programa multiplica dos números enteros sin usar el operador \*:

m = int(raw\_input())

n = int(raw\_input())

p = 0

**while** m > 0:

m = m - 1

p = p + n

**print** 'El producto de m y n es', p