Instrucciones condicionales

Las instrucciones condicionales permiten elegir una entre dos vías para resolver un problema, según sea el resultado de evaluar un predicado.

```
Si predicado_se_cumple:
    hacer esto
si no:
    hacer lo otro
```

Por ejemplo, para saber si alguien aprueba o no, basta con ver si la nota es un cinco, al menos:

```
Si "la nota es un cinco al menos":
    la respuesta es que sí, que el estudiante aprueba
si no:
    la respuesta es negativa, pues el estudiante no aprueba
```

La respuesta a la pregunta de si "la nota es un cinco al menos" a un sí o un no, esto es, True o False, que son los valores booleanos. En la instrucción condicional, siempre hay una expresión booleana que gobierna cuál es la instrucción que se realiza.

Instrucción condicional completa

Lo anterior se puede poner en Python así:

In [1]:

```
def aprobado(nota):
    if nota >= 5.0:
        return True
    else:
        return False

aprobado(3.0), aprobado(7.5), aprobado(5.0), aprobado(10.0), aprobado(4.999)
```

Out[1]:

```
(False, True, True, True, False)
```

La función anterior está pobremente documentada, y queremos progresar tan rápido que a menudo omitimos la documentación. Hela aquí de nuevo con una documentación más completa:

In [2]:

```
def aprobado(nota):
    """
    Averigua si una nota numérica es suficiente para aprobar o no

    Parameters
    ----------
    nota : float
        La nota numérica

    Returns
    -------
    bool
        True, si la nota es suficiente para aprobar
        False, si la nota no lo es

    Precondition
    ------------
    nota >= 0 and nota <= 10
    """
    if nota >= 5.0:
        return True
    else:
        return False

aprobado(3.0), aprobado(7.5), aprobado(5.0), aprobado(10.0), aprobado(4.999)
```

Out[2]:

```
(False, True, True, True, False)
```

Instrucción condicional incompleta

La rama "else" no siempre es necesaria:

In [3]:

```
def valor_absoluto(x):
    if x < 0:
      # cambio de signo del dato de entrada
      x = -x
    return x

valor_absoluto(10), valor_absoluto(-10)
```

Out[3]:

```
(10, 10)
```

Seguro que puedes proponer tú una documentación más completa de la función anterior. En este caso, podríamos haber usado también la instrucción condicional completa:

In [4]:

```
def valor_absoluto(x):
    if x < 0:
        return -x
    else:
        return x

print(valor_absoluto(10), valor_absoluto(-10))

# La misma función con su documentación completa:

def valor_absoluto(x):
    """
    Devuelve el valor absoluco de un número, real

    Parameters
    ----------
    x : float

    Returns
    -------
    float
        El valor absoluto de x
    """
    if x < 0:
        return -x
    else:
        return x

print(valor_absoluto(7), valor_absoluto(-7))
```

```
10 10
7 7
```

Otro ejemplo:

In [5]:

```
def max_min(x, y):
    if x < y :
        max = y
        min = x
    else:
        max = x
        min = y
    return max, min

print(max_min(2, 3), max_min(3, 2))
```

```
(3, 2) (3, 2)
```

Podemos anidar condicionales

In [6]:

```
def grado_poli(a, b, c):
    """
    Devuelve el grado del polinomio a*x^2 + b*x + c

    Parameters
    ----------
    a, b, c : float
        Coeficientes del polinomio

    Returns
    -------
    int
        Grado del polinomio
    """
    if a == 0:
        if b == 0:
            grado = 0
        else:
            grado = 1
    else:
        grado = 2
    return grado

grado_poli(2, 3, 4), grado_poli(0, 1, 2), grado_poli(1, 0, 0), grado_poli(0, 0, 1), grado_p
```

Out[6]:

```
(2, 1, 2, 0, 0)
```

Elecciones múltiples

In [7]:

```
def calificacion(num_nota):
    """
    Converierte una calificación numérica en su denominación

    Parameters
    ----------
    num_nota : float
        la nota numérica

    Returns
    -------
    str
        la denominación de la calificación

    Precondition
    ------------
    num_nota >= 0 and num_nota <= 10
    """
    if num_nota < 5:
        return "Suspenso"
    elif num_nota < 7:
        return "Aprobado"
    elif num_nota < 9:
        return "Notable"
    else:
        return "Sobresaliente"

calificacion(10), calificacion(1.23), calificacion(3.45), calificacion(4.999), calificacior
```

Out[7]:

```
('Sobresaliente', 'Suspenso', 'Suspenso', 'Suspenso', 'Notable', 'Aprobado')
```

Otro ejemplo más complicado

In [8]:

```
def resuelve_sistema_lineal(a, b, c, d, e, f):
    """
    Resuelve el sistema de ecuaciones lineales siguiente:
         a*x + b*y = e
         c*x + d*y = f

    Parameters
    ----------
    a, b, c, d, e, f : float
        Coeficientes de las ecuaciones

    Returns
    -------
    (float, float)
        Par de valores (x, y) que verifican ese sistema de ecuaciones

    Precondition
    ------------
      a*d - b*c != 0
    """
    if a == 0:
        y = e / b
        x = (f - d*y) / c
    else:
        y = (a*f - c*e)/(a*d - c*b)
        x = (e - b*y) / a
    return x, y

print(resuelve_sistema_lineal(1.0, 1.0, 0.0, 1.0, 2.0, 1.0))
print(resuelve_sistema_lineal(7.0, 2.0, 5.0, -1.0, 11.0, 3.0))
```

```
(1.0, 1.0)
(1.0, 2.0)
```

Errores comunes

No cubrir todos los casos posibles.

In [9]:

```
def val_absoluto_mal(x):
    if x < 0:
        return -x
    if x > 0:
        return x

val_absoluto_mal(0) == 0
```

Out[9]:

```
False
```

Por tanto, es preferible, para las elecciones múltiples, usar "if .... elif ... elif ... else"

Referencias

He aquí nuestra referencia, tomada de w3shhols:

```
https://www.w3schools.com/python/python_conditions.asp
```