

Actividad II - Condicionales

Exactas Programa

Invierno 2019

Esta serie de actividades están orientadas para practicar los contenidos vistos en la clase de introducción, específicamente la utilización de los condicionales.

No copies y pegues el ítem anterior, escribí desde cero cada ejercicio. La idea es que aprendas a escribir en Python.

Cada ejercicio que hagas, probalo en el **Python Tutor** (<http://pythontutor.com>).

Condicionales: If (si), Else (sino), Elif (sino si)

La instrucción If ejecuta un *camino* o el otro de acuerdo a si una condición es verdadera (**true**) o falsa (**false**).

```
if <condicion>:
    <instrucciones para caso verdadero>
else:
    <instrucciones para caso false>
<instrucciones despues de if>
```

En Python podría verse algo así (se supone que la variable **myA** está previamente definida con algún valor):

```
1 if myA==2:
2     myA=myA+4
3 else:
4     myA = 0
5 myB=2*myA
```

En este ejemplo, se ejecutará la instrucción de la línea 2 siempre que **myA** tenga el valor 2, en tanto que se ejecutará la línea 4 en otro caso. La línea 5 se ejecutará en cualquiera de los dos casos, tanto si la condición **myA==2** dio **true** o dio **false**.

Ejercicios para hacer

1. Completar y probar:

```
def decir_si_es_mas_grande_que_5():
    numero = <completar>
    return numero > 5
```

2. Completar y probar:

```
def decir_si_la_longitud_es_mayor_a_5():
    unNombre = "Camel Black"
    return <completar - una sola linea>
```

3. Completar y probar:

```
def decir_si_es_mas_grande_que(unNumero):
    if <completar - son varias lineas>
    return <completar>
```

4. Completar y probar:

```
def decir_si_es_igual_a(unNumero):  
    numero = 10  
    return <completar>
```

5. Completar y probar:

```
def decir_si_la_longitud_es_igual_a(unNombre, unNumero):  
    return <completar>
```

Condiciones combinadas

El `if` *evalúa* la expresión condicional para saber cuál camino tomar. Esta expresión debe *devolver* un valor que sea verdadero o falso (una condición válida es `a==4`). Sin embargo, las condiciones no tienen por qué ser comparaciones directas sobre una única variable, se pueden escribir expresiones tan complicadas como uno quiera.

```
1 def sumo_uno_si_digito(numero):  
2     if numero>=0 and numero<10:  
3         resultado = numero+1  
4     return resultado  
5 res = sumo_uno_si_digito(4)  
6 print(res)  
7 res = sumo_uno_si_digito(14)  
8 print(res)
```

En este programa, se define una función que utiliza un `if` que solo suma si se cumplen que el número recibido es, a la vez, mayor o igual a 0 y menor 10 (es decir, es un dígito decimal). El valor asignado a `res` en la línea 5 será 5, en tanto que el valor asignado a la misma variable en la línea 7 será 14.

Los operadores lógicos que vamos a usar más habitualmente son:

- **and**: devuelve verdadero solo si **las dos** condiciones que lo encierran (que están alrededor) son verdaderas. Por ejemplo, la expresión `2>1 and 2>0` tiene como resultado `true`.
- **or**: devuelve verdadero si **alguna** de las dos condiciones que lo encierran es verdadero. Por ejemplo, la expresión `2<10 or 2>0` dará `true` ya que, a pesar de que la primera condición es falsa, la segunda es verdadera.

Las condiciones se pueden ir combinando y formando expresiones tan complejas como uno quiera. A esta altura del curso, no vamos a profundizar en esto y nos quedaremos en la combinación de un par de expresiones únicamente.

Ejercicios para hacer

En algunos de los siguientes ejercicios, usaremos la función `es_par` que indica si el número recibido es múltiplo de 2:

```
def es_par(unNumero):  
    resto = unNumero % 2  
    return resto == 0
```

Completar y probar las siguientes funciones:

```
1. def devolver_valor_mas_grande(valor1, valor2):  
    if valor1 > valor2:  
        resultado = <completar>  
    else:  
        resultado = <completar>  
    return resultado
```

```

2. def devolver_el_doble_si_es_par(unNumero):
    if es_par(unNumero):
        <completar>
    else:
        <completar>
    return resultado
3. def devolver_segun_condiciones_locas(unNumero):
    if (unNumero == 2):
        <completar +1>
    elif (unNumero <= 10):
        <completar para que devuelva el doble>
    elif <completar numero entre 20 y 34>:
        <completar + 5>
    else:
        <completar, da cero>
    return resultado

```

Más ejercicios para practicar

Definir e implementar las siguientes funciones, en todos los casos se recomienda escribirlas y probarlas con distintos ejemplos para verificar su funcionamiento.

1. Definir una función que recibe un número y devuelve el doble si es más grande que 10 y menor a 20
2. Definir una función que recibe un número y devuelve “Rayos y Centellas” si es más grande que 20 o menor a 5
3. Definir una función que recibe un número y devuelve “Está en el rango deseado” si el valor está entre 5 y 10 y “Fuera de Rango” en caso contrario
4. Definir una función que recibe un número y devuelve “Menor a 5” si el valor es menor a 5, “Entre 10 y 20” si el valor está entre 10 y 20, y en cualquier otro caso que devuelva “Número muy grande”