

Introducción a la Programación

Práctica 2 – Condicionales

Versión del 1 de agosto de 2022

El *camino* del programa

Todos los programas que vimos hasta ahora hacen exactamente lo mismo cada vez que se ejecutan. Esto es, empiezan por la primera línea, y van hacia abajo ejecutando todas las líneas de a una por vez. A veces, sin embargo, no se quiere que el programa ejecute todas las líneas, sino que se puedan tomar decisiones automáticamente, ejecutando ciertas partes sólo en determinados casos.

Cláusula if

Supongamos que hicimos un programa que permite al usuario intentar resolver una adivinanza:

```
respuesta = input("Dos niñas van a la par pero no se pueden mirar.")
print("Su respuesta es ", respuesta)
```

Este programa tiene una novedad: cuando usa el comando **print**, lleva entre paréntesis dos cadenas separadas por coma. La primera está escrita explícitamente, la otra está guardada en una variable. El resultado de esto, como es de esperarse, es que se mostrarán ambas cadenas una seguido de la otra.

El problema de este programa es que no sabe decidir si la respuesta ingresada por el usuario es la correcta. Hasta ahora no tenemos forma de hacerlo, veamos como serían los pasos si el programa se escribiera en castellano:

1. Pedir al usuario que ingrese una respuesta para la adivinanza.
2. Si la respuesta es "las pupilas", entonces mostrar "Felicitaciones, adivinaste!"

Pasar la primera línea a Python es sencillo, y ya sabemos como hacerlo. La línea 2, en cambio, por ahora no tenemos forma de escribirla. Para ello vamos a presentar a la estructura si, que usaremos dicho en inglés (if). Veamos cómo se usa escribiendo el programa de nuevo:

```
respuesta = input("Dos niñas van a la par pero no se pueden mirar.")
if respuesta == "las pupilas":
    print("Felicitaciones, adivinaste!")
print("Gracias por participar.")
```

al ejecutar el programa se comenzará como de costumbre por la primera línea, pidiendo al usuario que ingrese la respuesta. Lo más interesante es lo que pasa justo después, entre las líneas dos y tres. En esas líneas tenemos la condición, y lo que se ejecutará si ella es verdadera. Cuando queremos que una línea del programa se ejecute solamente si se cumple cierta condición, escribiremos esa línea dentro de un **if**, en este caso:

```
if respuesta == "las pupilas":
    print("Felicitaciones, adivinaste!")
print("Gracias por participar.")
```

Lo que acabamos de ver es un condicional, también conocido como una cláusula *si*, o un *if*¹. Lo interesante de este programa es que no hace siempre lo mismo, sino que cambia su *camino* en función de la nota que se le ingresa. En el caso de que la respuesta ingresada sea “las pupilas”, se ejecutan todas las líneas que estén escritas con sangría debajo del *if*, es decir,

```
print("Felicitaciones, adivinaste!")
```

En el caso de que la respuesta ingresada sea distinta de “las pupilas”, se saltea todo lo que está escrito con sangría después del *if*, sin ejecutar nada al pasar, y luego se sigue ejecutando por la línea que sigue alineada al *if*, en este caso:

```
print("Gracias por participar.")
```

Algo que cabe remarcar, es el uso del operador `==`. Este operador sirve para comparar dos expresiones y determinar si son iguales. (No confundirlo con el operador de asignación `=`).

En general, la cláusula *if* se escribe siguiendo este esquema:

```
if condicion_a_cumplir :  
    ... código a ejecutar si la condición se cumple ...
```

Donde la *condicion_a_cumplir* es una *proposición* cualquiera, es decir, una expresión que puede ser verdadera o falsa.

Todo el código que está dentro del *if* debe alinearse más a la derecha que el que está afuera, dejando una sangría o indentación². Esto hará que sea visualmente más sencillo leer y entender un programa.

Cláusula *else*

Hasta ahora nuestro programa solo informa que el usuario adivinó. Sería bueno que también informe al usuario si su respuesta no fue correcta. Esto se puede hacer de la siguiente manera:

```
respuesta = input("Dos niñas van a la par pero no se pueden mirar.")  
if (respuesta == "las pupilas") :  
    print("Felicitaciones, adivinaste!")  
else:  
    print("Ay qué lástima. Perdiste.")  
print("Gracias por participar.")
```

En este caso, decimos que dentro del *if* tenemos dos *ramas*: la afirmativa, que es la que se ejecutará si la condición es verdadera

```
mostrar("Felicitaciones, adivinaste!")
```

y la negativa, que se ejecutará en el caso de que la condición sea falsa

```
mostrar("Ay qué lástima. Perdiste.")
```

Lógicamente, para cualquier respuesta que ingrese el usuario, siempre se ejecutará una y sólo una de las dos partes del condicional. La otra parte se pasará de largo sin ejecutarse. Luego

¹del inglés *si*

de ejecutar la rama correspondiente, sin importar si fue la afirmativa o la negativa, se continúa ejecutando a partir de la línea que sigue al final del bloque if, es decir, al código que por estar alineado (sin sangría) al mismo nivel que la condición, no es parte del condicional. Eso es:

```
mostrar("Gracias por participar.")
```

El condicional es una estructura que nos permite realizar programas que cambien su camino de ejecución según decisiones tomadas en el mismo momento en que se ejecuta el programa. Es interesante comentar que todo el código que uno escribe dentro de las ramas de un if (negativa o afirmativa) se rige por las mismas reglas que vimos hasta ahora, es decir, allí se pueden incluir asignaciones de variables, el uso de input y print, y por qué no también, otros if. Esto último se conoce como condicionales anidados. En estos casos, las sangrías que se utilicen serán siempre relativas a la ubicación de la palabra clave **if**.

La lógica y la computación

La lógica constituye un recurso básico en toda forma de razonamiento y fundamentación del conocimiento, siendo el soporte de toda actividad científica. Dentro de la rama de la computación, la lógica nos permite construir programas que realicen distintas tareas en función de las posibilidades que se presenten a la hora de ejecutarlos.

Valores lógicos

Los valores lógicos (o valores Booleanos) son 2: Verdadero y Falso. En programación se los conoce por sus nombres en inglés **True** y **False**. Los llamamos valores, porque hay expresiones que evalúan mediante reducciones a uno de estos valores. Esas expresiones son entonces llamadas **expresiones lógicas**. Cabe aclarar que los valores lógicos **True** y **False** también pueden asignarse a variables.

Las expresiones lógicas, según los valores de las variables que en ellas aparezcan, pueden ser verdaderas o falsas. Es por esto que se las suele llamar *proposiciones* o *condiciones*. (Utilizadas en estructuras if, se las suele llamar *guardas*).

Un ejemplo de una expresión lógica es:

```
x == "las pupilas"
```

Esta expresión reducirá en el valor **True** si la variable *x* guarda la cadena "las pupilas", y reducirá en el valor **False** en cualquiera otro caso. El encargado de esto es el operador **==**, que compara los valores a los que reducen las expresiones que están a cada lado.

Operadores lógicos

Si quisieramos permitir que la cadena esté escrita completamente en mayúsculas, podríamos tener una expresión que sea verdadera tanto para la cadena "las pupilas" o "LAS PUPILAS". Para eso utilizaremos el operador lógico **OR**. Entonces dicha expresión sería:

```
x == "las pupilas" OR x == "LAS PUPILAS"
```

De este modo, podemos combinar expresiones lógicas para formar expresiones más complejas utilizando los operadores lógicos: **AND**, **OR** y **NOT**. **AND** es la conjunción (y), **OR** es la disyunción (o), **NOT** es la negación (no).

Además de estos operadores, existen operadores de comparación (a veces llamados operadores relacionales) que trabajan con expresiones matemáticas y reducen a Verdadero o Falso. Las condiciones del if pueden contener cualquiera de estos operadores:

- `==` (es igual a)
- `!=` (es distinto que)
- `<` (menor que)
- `>` (mayor que)
- `<=` (menor o igual que, \leq)
- `>=` (mayor o igual que, \geq)

Ejercicios resueltos de ejemplo

Ejercicio de ejemplo 1

Hacer en papel y luego en Python un programa que pida un entero entre 1 y 10, utilizando el mensaje “Adiviná en qué número estoy pensando!”, y muestre “Adivinaste” si el usuario ingresa el número 7.

Resolución

En este ejercicio nos piden que hagamos un programa que pida al usuario que ingrese un número entre 1 y 10, y en caso de que sea igual a 7, muestre un mensaje, y en caso de que el número ingresado sea distinto de 7, muestre otra cosa.

Pseudocódigo:

```
x = int(input("Adiviná en qué número entero del 1 al 10 estoy pensando!"))
if x == 7 :
    | print("Adivinaste")
```

Ejercicio de ejemplo 2

Modificar el programa del ejercicio anterior para que en caso de que el usuario no adivine el número, muestre por pantalla el mensaje “Perdiste”.

Resolución:

```
x = int(input("Adiviná en qué número estoy pensando!"))
if x == 7 :
    | print("Adivinaste")
else:
    | print("Perdiste")
```

Ejercicio de ejemplo 3

Modificar el programa del ejercicio anterior para que en caso de que el usuario ingrese un valor menor a 7 muestre “Te quedaste corto” y en caso de ingresar algo mayor a 7 muestre . “Te pasaste”.

Resolución:

```
x = int(input("Adiviná en qué número estoy pensando!"))
if x == 7 :
    | print("Adivinaste")
else:
    | if x < 7 :
    |   | print("Te quedaste corto")
    | else:
    |   | print("Te pasaste")
```

Ejercicio

Para terminar, probar el último ejercicio de ejemplo en Python y contestar las siguientes preguntas:

1. ¿Qué se muestra si el usuario ingresa el número 0?
2. ¿Qué se muestra si el usuario ingresa el número 7?
3. ¿Qué se muestra si el usuario ingresa el número 8?
4. ¿Son esos resultados los que esperábamos?

Notas preliminares

- Los ejercicios marcados con el símbolo ★ constituyen un subconjunto mínimo de ejercitación. Sin embargo, aconsejamos fuertemente hacer todos los ejercicios. Escribir todos los programas en pseudocódigo en papel y luego pasarlos a python y probarlos en la computadora

Ejercicio 1

Este programa chequea una serie de condiciones para los tres valores ingresados por el usuario. Correrlo tal cual está en Python. Luego reemplazar donde dice **True** por una expresión lógica que sea **True** o **False** según corresponda, en lugar de siempre **True** como ahora.

```
a = int(input("Ingrese un numero entero"))
b = int(input("Ingrese un numero entero"))
c = int(input("Ingrese un numero entero"))
print("Usted ingresó los valores:", a, b, c)
print(a, "es mayor que", b, True)
print(a, "y", b, "son iguales", True)
print(a, "es el mayor de todos", True)
print(b, "es el menor de todos", True)
print(a, "es mayor que alguno de los otros dos", True)
print(a, "es menor o igual que alguno de los otros dos", True)
print("Los tres numeros son iguales", True)
print("Los tres numeros son distintos", True)
print(a, "es par", True)
print("alguno es par", True)
print("ninguno es par", True)
print("todos son pares", True)
print(a, "es multiplo de 3", True)
print(a, "es multiplo de 3 y de 5", True)
print(a, "es multiplo de 3 y par", True)
print(a, "-", b, "da un numero positivo", True)
print(a, "-", b, "da un numero par positivo", True)
```

Ejercicio 2

Un ciudadano argentino está exento de votar en estos casos:

- Tiene más de 70 años
- Tiene entre 18 y 70 años pero se encuentra a más de 500 km del centro de votación.

Suponiendo que las variables *edad* y *distancia* representan la edad y la distancia del ciudadano, escribir la expresión lógica que representa esta situación.

Ejercicio 3

Para cada uno de los siguientes programas indicar qué se imprime cuando se ejecuta

a)

```
a = 10
if a ≠ 0 :
    print("perro")
```

b)

```
a = 10
if a > 0 :
    print("manzana")
else:
    print("naranja")
```

c)

```
a = 10
if a > 0 :
    print("Te quiero")
print("bien lejos.")
```

d)

```
a = 5
b = 3
c = 2
if a < b * c :
    print("Hola!")
else:
    print("Chau!")
```

e)

```
p1 = 3,14
p2 = 3,141569
if p1 == p2 :
    print("p1 y p2 son iguales!")
else:
    print("p1 y p2 no son iguales!")
```

f)

```
a = "Hola"
b = "hola"
if a == b :
    print("Python es insensible!")
else:
    print("Python es muy sensible!")
```


Ejercicio 4

Leer el siguiente programa. ¿Para qué valores de la variable a imprime “hola!” cuando se ejecuta? ¿Y para cuáles “chau!”?

```
a = int(input("Ingrese un valor para a"))
if a == 0 :
    | print("hola!")
else:
    | print("chau!")
```

Ejercicio 5

Escribir en papel un programa que pida una nota y que en el caso de que sea menor a cuatro muestre “Debe recuperar”. Luego pasarlo a Python.

Ejercicio 6

- a) Escribir en papel un programa que pida al usuario dos números, y que muestre en pantalla al mayor de ambos. Luego hacer 3 corridas de escritorio, luego pasarlo a Python y por último correr el programa con los valores iniciales de las corridas y verificar que funciona como se esperaba.
- b) Ídem anterior pero para encontrar el menor

Ejercicio 7

Escribir en papel un programa que pida al usuario dos números de punto flotante y muestre su promedio. Además, si el promedio es mayor que 7 el programa debe mostrar en pantalla “Aprobado” y si no, debe mostrar “Desaprobado”. Después de hacerlo en papel, pasarlo a Python.

Ejercicio 8

Escribir en papel un programa que tome un número entero positivo ingresado por el usuario y muestre por pantalla “Usted ingresó un número de una sola cifra” o “Usted ingresó un número de más de una cifra” según corresponda. Realizar 4 corridas de escritorio, escribirlo en Python y luego correrlo en la computadora con los valores iniciales de las corridas y verificar que hayan dado como se esperaba.

Ejercicio 9 ★

Se tiene la siguiente lista con DNIs de personas.

- 30612453
- 23763290
- 21448503
- 34582048
- 15364857

Dado otro número de DNI cualquiera, se desea construir un programa que determine si es alguno de los existentes en el listado. Escribir el programa en papel y luego pasarlo a Python.

Ejercicio 10 ★

Hacer en pseudocódigo y luego un programa que calcule el importe que se le facturará a un cliente por consumo de electricidad sabiendo que la compañía que se la provee cobra una tarifa fija de 480 pesos que incluye los primeros 200 KW consumidos y los KW excedentes se los cobra a 25,5 pesos el KW, además se agregan \$78 de impuestos. Se leen los valores del medidor al comienzo y al fin del período.

Ejercicio 11 ★

Se desea escribir un programa que pida al usuario tres números y luego muestre el mayor de ellos. Escribir el programa en papel, realizar 3 pruebas de escritorio y luego pasarlo a Python y verificar los resultados.

Ejercicio 12

Un profesor clasifica las notas de sus alumnos de la siguiente manera:

1-3	Reprobado
4-6	Debe rendir examen final
7-10	Eximido

- a) Escribir un programa que indique la clasificación de una nota.
- b) Escribir un programa que pida 3 notas, calcule el promedio e indique la clasificación del mismo.

Ejercicio 13 ★

Escribir un programa que pida al usuario dos enteros y que luego muestre si el primero es mayor que el segundo o viceversa.

Ejercicio 14 ★

Escribir un programa que pida al usuario dos enteros y los guarde en dos variables. Si el primero de los valores fuera menor que el segundo, el programa deberá además intercambiar los valores de las variables y mostrarlos de mayor a menor.

Ejercicio 15 ★ ♣

Escribir un programa que pida al usuario tres enteros y los guarde en tres variables a , b y c . El programa deberá luego hacer que en la variable a quede el menor de los valores recibidos, en b el intermedio y en c el mayor (es decir, ordenará los valores).

Ejercicio 16 ★

Un año es bisiesto si es múltiplo de 4. Pero no siempre, las excepciones son los años múltiplos de 100 que no son múltiplos de 400 (1900 no es bisiesto pero 2000, sí). Escribir en papel un programa que diga si un año ingresado por el usuario es bisiesto, realizar varias pruebas de escritorio, luego pasarlo a Python y verificar los resultados.

Ejercicio 17 ★ ♣

Escribe un programa que pida los coeficientes de una ecuación de primer grado ($ax + b = 0$) y escriba la solución. Recuerda que una ecuación de primer grado puede no tener solución, tener una solución única, o que todos los números reales sean solución.

Ejercicio 18 ★ ♣

Una función cuadrática se escribe como $ax^2 + bx + c$. La misma puede tener una, dos o ninguna raíz. Escribir un programa que pida al usuario los datos de la misma, es decir, a , b y c , y muestre todas sus raíces, o el mensaje “No tiene raíces” cuando corresponda. Recordar que las raíces están dadas por la fórmula

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