

# Condicionales / Decisiones

---

## Temas incluidos en la guía

- Python's basics
- Funciones
- Condicionales

## Ejercicios

Nota: (☆☆☆, ★☆☆, ★★★) Esta notación indica la dificultad (ascendente).

1. ★☆☆ Corrija el siguiente programa defectuoso para que imprima la negación de la variable lógica `state`.

```
state = True
print(state)
```

2. ★☆☆ Corrija este programa defectuoso que produce un error al intentar ejecutarlo.

```
cake = True
if cake = True:
    print("cake es True")
else:
    print("The cake is a lie")
```

3. ★☆☆ Escriba una función que diga si un número es par o impar.

```
def is_even(number: int) -> bool:
    # complete me
```

4. ★☆☆ Escriba una función que dados dos números `a` y `b` compare estos números y retorne otro número indicando el resultado de la comparación del siguiente modo:

- 0, si `a` y `b` son iguales,
- negativo, si `a` es menor que `b`,
- positivo, si `a` es mayor que `b`.

```
def cmp_number(a: float, b: float) -> float:
    # complete me
```

5. ★★☆☆ Escriba una función que pida un número  $x$ , luego pida otro  $y$ , y devuelva los dos números multiplicados y divididos. En la consola debe imprimirse lo siguiente:

```
Ingrese el primer número:  x
Ingrese el segundo número: y

El resultado de la multiplicación es x * y = z
El resultado de la división es x / y = w
```

Note que  $x$  e  $y$  están alineados a la misma distancia, es decir que están separados por un tab y no por espacios.

- ¿Si  $x=1$  e  $y=10$ , la división devuelve 0.1? (vea la diferencia entre int y float)
- ¿Qué pasa con la división si  $y=0$ ? Agregue una verificación que se fije que  $y$  no sea 0. En caso de que sea 0, que no haga ningún cálculo e imprima:

```
No se puede dividir por 0
```

6. ★★☆☆ Implemente una función que reciba tres variables booleanas ( $a, b, c$ ) y devuelva otra variable booleana  $x$  según la tabla de verdad que se indica abajo. Luego, llamar a esa función e imprimir en pantalla la tabla completa para los distintos valores de  $a$ ,  $b$  y  $c$  (como se muestra a continuación).

| a | b | c | x |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

7. ★★☆☆ Escriba una función que calcule y retorne las raíces de la ecuación:  $ax^2 + bx + c = 0$ . Recuerde que las raíces se obtienen como:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Sólo debe resolver la ecuación para el caso en que el discriminante ( $b^2 - 4ac$ ) es mayor que 0.

8. ★★★ Escribir un programa que dado un día del año (1 a 366) ingresado por el usuario, imprima el día de la semana que le corresponde. Debe asumir que el año comenzó, por ejemplo, un domingo. Por

ejemplo: si se ingresa '5', imprime 'jueves', si se ingresa '10' imprime 'martes', si se ingresa '294' imprime 'sabado'.

9. ★★★ Escribir una función que dadas dos rectas definidas por su pendiente y su ordenada al origen devuelva la abscisa en la que se intersectan. Validar lo que considere necesario.

10. ★★★ Escribir funciones que resuelvan los siguientes problemas:

- Dado un año indicar si es bisiesto.

**Nota:** Un año es bisiesto si es un número divisible por 4, pero no si es divisible por 100, excepto que también sea divisible por 400.

- Dado un mes y un año, devolver la cantidad de días correspondientes (por ejemplo: 2022 y 11 devuelve 30).
- Dada una fecha (día, mes, año), indicar si es válida o no.
- Dada una fecha, indicar los días que faltan hasta fin de mes.
- Dada una fecha, indicar los días que faltan hasta fin de año.
- Dada una fecha, indicar la cantidad de días transcurridos en ese año hasta esa fecha.
- Dadas dos fechas (día1, mes1, año1, día2, mes2, año2), indicar el tiempo transcurrido entre ambas, en años, meses y días.

**Nota:** reutilizar las funciones desarrolladas en cada punto.

11. ★★★ Escribir un programa que inicie pidiendo al usuario que ingrese la fecha e imprima cuántos días faltan para la primavera. Por ejemplo:

- si el usuario ingresa el 2021-07-18 (18 de julio de 2021), faltan 65 días,
- si ingresa el 2021-01-01 (1 de enero de 2021), faltan 263,
- si ingresa el 2020-01-01 (1 de enero de 2020), faltan 264,
- y si ingresa el 13 de noviembre de 2021, faltan 312 días.

12. ★★★ Queremos conocer la ganancia que nos da un plazo fijo a tasa nominal anual (TNA) constante, para ello tenemos que escribir un programa que nos solicite la TNA, el capital fijo, la fecha actual y la fecha en la que queremos recuperar lo depositado. Con esos datos, obtener el capital final al terminar el plazo. Tenga en cuenta que **no** se aceptan plazos fijos por menos de 30 días.

13. ★★★ Dado un valor de TNA, queremos saber el tiempo que debemos mantener depositado un capital en un plazo fijo para obtener una ganancia dada. Escriba un program que nos resuelva dicho problema.

14. ★★★ Dado un valor de TNA, queremos saber qué capital debemos depositar por un tiempo determinado para obtener una ganancia predefinida. Escriba un program que nos resuelva dicho problema.

15. ★★★ A la fecha, un plazo fijo en TOR tiene una TNA del 42% para depósitos menores a 10M\$, y 39.5% para depósitos mayores. Mientras tanto, un plazo fijo en XFG tiene una TNA fija del 0.5%. El

tipo de cambio entre las monedas es de 200 XFG por 1 TOR, y aumenta diariamente un 0.045% (el día 0 vale 200, luego de  $n$  días vale  $200 \cdot (1 + 0.045/100)^n$ ).

- Escribir un programa que, dados los datos del enunciado, un capital y un período para un plazo fijo solicitados al usuario, me indique cuál de las siguientes opciones me conviene:
  - Hacer un depósito en TOR, o
  - Convertir a XFG, y hacer el depósito, y convertir el monto final a XFG.