

Condiciones booleanas más complejas

Ejercicio 1

En Renfe te encargan una función para determinar el precios de los billetes con la siguiente promoción:

- El billete de ida y vuelta a cualquier ciudad de España cuesta €0.1 por Km recorrido en el primer tramo.
- Si la distancia es superior a 600 Km, el precio por Km baja a €0.08.
- Si la distancia es inferior a 100 km, el coste por km es de €0.17
- Si la vuelta es 7 días o más después de la ida, el precio total tiene un descuento del 12%.

Crea la función `ticket_cost` que recibe: distancia en km al destino final y número de días entre la ida y la vuelta. Debe de devolver el coste total.

- ¿Es una función total o parcial?

Ejercicio 2

En Naciones Unidas, para determinar el sueldo de un empleado, se clasifican los puestos con una letra y un número (G1, G2, P1,P2,P3, P4, P5, P6, D1, D2). Las letras se corresponden a "General", "Profesional" y "Directivo".

Se van a actualizar los sueldos de los de nivel P. Se te encarga que escribas una función `salary_rise` que devuelva el porcentaje que le corresponde a cada uno de los P, siguiendo los datos de abajo:

- P1: 20%
- P2: 22%
- P3: 21%
- P4: 19%
- P5: 24%
- P6: 15%

la función recibe le número (sin la p) y devuelve el porcentaje que corresponde.

- ¿Es una función total o parcial?

Ejercicio 3

Crea una función `bigger` que recibe dos números.

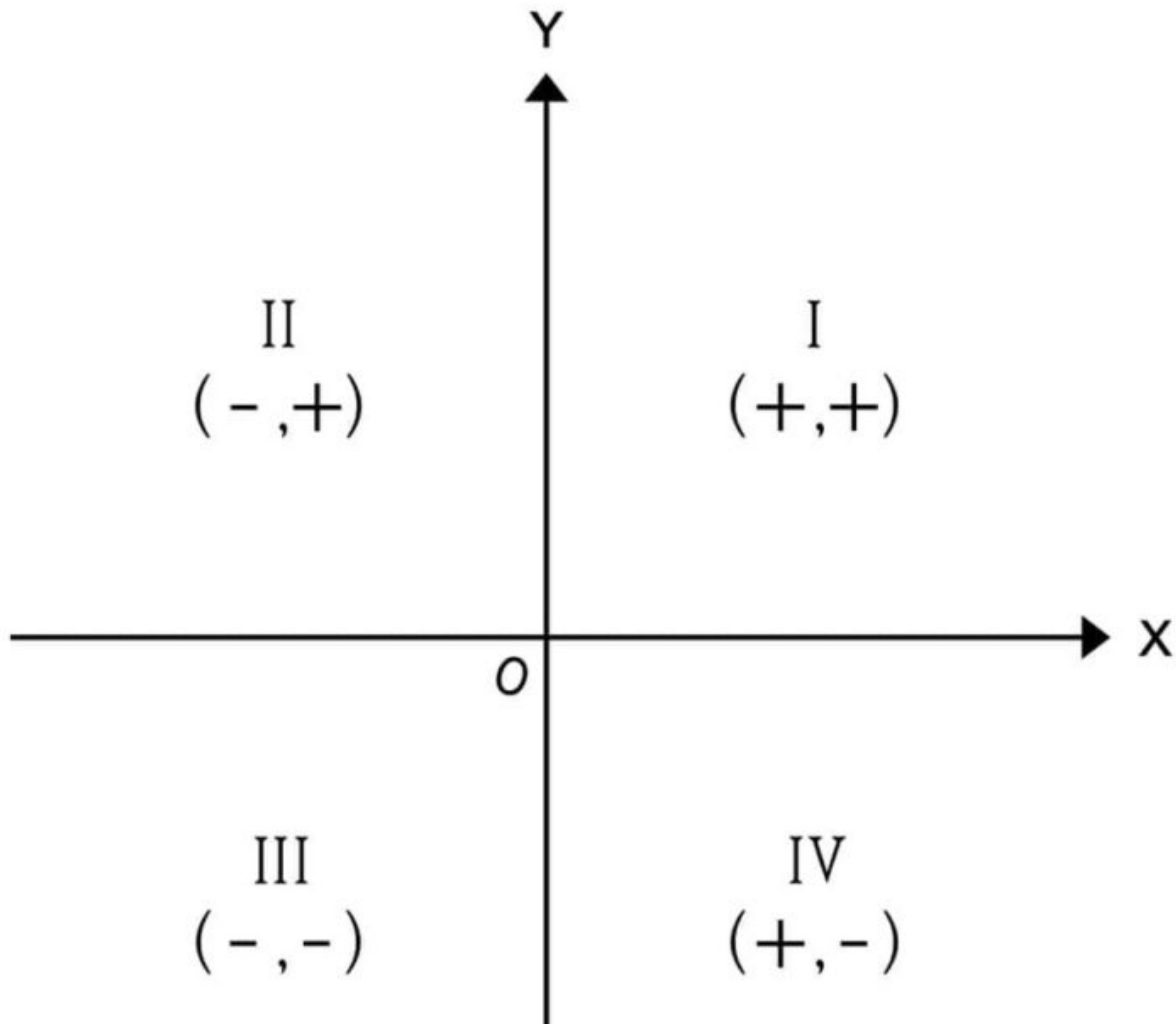
- Si son iguales, devuelve cualquiera de ellos
- Sino, devuelve el que sea mayor.

Ejercicio 4

Crea una función que recibe la calificación de un estudiante (entre 0 y 100).

Devuelve la clasificación correspondiente: "Sobresaliente" (90-100), "Notable" (80-89), "Aprobado" (65-79) o "Reprobado" (0-64).

Ejercicio 5



Crea la función `classify` que recibe las coordenadas de un punto (x, y) en el plano cartesiano.

Devuelve una cadena que indica en qué cuadrante se encuentra el punto ("primer", "segundo", "tercer" o "cuarto cuadrante") o si está sobre alguno de los ejes ("eje x", "eje y"), o por último, si está en el origen ("origen").

- ¿Es una función parcial o total?

Ejercicio 6

Crea la función `is_leap` que recibe un año y determina si un año es bisiesto.

Un año es bisiesto si es divisible entre 4, excepto aquellos divisibles entre 100 pero no entre 400.

- ¿Es un predicado?

Ejercicio 7

Función que recibe una edad y la clasifica de la siguiente manera:

- "Infancia" (0-12 años),
- "Adolescencia" (13-17 años),
- "Jueventud" (18-29 años),
- "Adulthood" (30-59 años)
- "Vejez" (60+ años)
- ¿Es una función parcial o total?
- ¿Es un predicado?

Ejercicio 8

¿Recuerdas la función `bmi` que recibía una altura y peso y te devolvía un índice?

Crea la función `classify_bmi` que recibe un `bmi` y devuelve una cadena indicando en qué rango está:

- "Delgado" : $bmi < 18.5$
- "Saludable": $18.5 < bmi < 24.9$
- "Gordito" : 25 a 29.9
- "Gordo" : 30 a 34.9
- "Obeso" : 35 a 39.9
- "Jabba the Hutt" : más de 40

Ejercicio 9

Ejercicios extra

Los he creado un amigo que luego os presentaré.

Recuerda utilizar las sentencias condicionales `if`, `else` y, si es necesario, `elif` en la implementación de las funciones para controlar el flujo del programa y devolver el resultado adecuado en función de los argumentos proporcionados.

Crea la función `days_in_month(month, year)` que acepta un número de mes (1-12) y un año, y devuelve la cantidad de días que tiene ese mes. Asegúrate de tener en cuenta si es un año bisiesto para febrero.

Crea la función `color_semaforo(color)` que acepta el color de la luz en un semáforo (rojo, amarillo o verde) y devuelve una cadena que indica qué acción debe tomar un conductor: detenerse, precaución o avanzar.

Crea la función `estacion_del_anho(mes)` que acepta un mes (1-12) y devuelve una cadena que indica la estación del año correspondiente en el hemisferio norte: primavera, verano, otoño o invierno.

Averigua por tu cuneta cuándo empieza cada estación.

Crea la función `rango_velocidad(velocidad)` que acepta una velocidad en km/h y devuelve una cadena que indica si la velocidad es "Lenta", "Moderada", "Rápida" o "Muy rápida".

Crea la función `tipo_triangulo(a, b, c)` que acepta las longitudes de los tres lados de un triángulo y devuelve una cadena que indica si el triángulo es equilátero, isósceles o escaleno.

Crea la función `es_vocal(letra)` que acepta un carácter y devuelve `True` si es una vocal (mayúscula o minúscula) y `False` si no lo es.

Un carácter es una cadena que sólo tiene una letra. Por ejemplo "u" es un carácter, pero "uau" no lo es.

Crea la función `calcular_precio_final(precio, impuesto, descuento)` que acepta el precio original de un producto, el porcentaje de impuesto y el porcentaje de descuento, y devuelve el precio final después de aplicar el impuesto y el descuento.

Crea la función `numero_a_palabras(numero)` que acepta un número entero entre 1 y 10 y devuelve una cadena con el número en palabras (por ejemplo, "uno", "dos", etc.). Si el número está fuera del rango, devuelve "Número fuera de rango".

Crea la función `multiplo_de(numero, base)` que acepta dos números enteros y devuelve `True` si el primer número es múltiplo del segundo, y `False` en caso contrario.

Crea la función `calcula_tarifa(edad)` que acepta la edad de un visitante y devuelve el precio de la entrada a un evento. Si el visitante es menor de 5 años, la entrada es gratuita; si tiene entre 5 y 18 años, el precio es de 10 dólares; y si tiene 19 años o más, el precio es de 15 dólares.

Crea la función `clasifica_numeros(a, b, c)` que acepta tres números enteros y devuelve una cadena que indica si los números están en orden ascendente, descendente o si no tienen un orden específico.

Crea la función `tiempo_a_segundos(horas, minutos, segundos)` que acepta horas, minutos y segundos y devuelve el tiempo total en segundos.

Crea la función `puntuacion_juego(puntuacion)` que acepta la puntuación en un juego y devuelve una cadena que indica el rango de puntuación: "Principiante" (0-500), "Intermedio" (501-2000) o "Experto" (más de 2000).