

- 4.3.1.6 LABORATORIO: Un año bisiesto: escribiendo tus propias funciones

LABORATORIO

Tiempo Estimado

10-15 minutos

Nivel de Dificultad

Fácil

Objetivos

Familiarizar al estudiante con:

- Proyectar y escribir funciones con parámetros.
- Utilizar la instrucción `return`.
- Probar las funciones.

Escenario

Tu tarea es escribir y probar una función que toma un argumento (un año) y devuelve `True` si el año es un *año bisiesto*, o `False` si no lo es.

Parte del esqueleto de la función ya está en el editor.

Nota: también hemos preparado un breve código de prueba, que puedes utilizar para probar tu función.

El código utiliza dos listas: una con los datos de prueba y la otra con los resultados esperados. El código te dirá si alguno de tus resultados no es válido.

- 4.3.1.7 LABORATORIO: ¿Cuántos días?: escribiendo y utilizando tus propias funciones

LABORATORIO

Tiempo Estimado

15-20 minutos

Nivel de Dificultad

Medio

Requisitos Previos

LABORATORIO 4.1.3.6

Objetivos

Familiarizar al estudiante con:

- Proyectar y escribir funciones parametrizadas.
- Utilizar la instrucción `return`.
- Utilizar las funciones propias del estudiante.

Escenario

Tu tarea es escribir y probar una función que toma dos argumentos (un año y un mes) y devuelve el número de días del mes/año dado (mientras que solo febrero es sensible al valor `year`, tu función debería ser universal).

La parte inicial de la función está lista. Ahora, haz que la función devuelva `None` si los argumentos no tienen sentido.

Por supuesto, puedes (y debes) utilizar la función previamente escrita y probada (LABORATORIO 4.1.3.6). Puede ser muy útil. Te recomendamos que utilices una lista con los meses. Puedes crearla dentro de la función; este truco acortará significativamente el código.

Hemos preparado un código de prueba. Amplíalo para incluir más casos de prueba.

- 4.3.1.8 LABORATORIO: Día del año: escribiendo y utilizando tus propias funciones

LABORATORIO

Tiempo Estimado

20-30 minutos

Nivel de Dificultad

Medio

Requisitos Previos

LABORATORIO 4.1.3.6

LABORATORIO 4.1.3.7

Objetivos

Familiarizar al estudiante con:

- Proyectar y escribir funciones con parámetros.
- Utilizar la sentencia `return`.
- Construir un conjunto de funciones de utilidad.
- Utilizar las funciones propias del estudiante.

Escenario

Tu tarea es escribir y probar una función que toma tres argumentos (un año, un mes y un día del mes) y devuelve el día correspondiente del año, o devuelve `None` si cualquiera de los argumentos no es válido.

Debes utilizar las funciones previamente escritas y probadas. Agrega algunos casos de prueba al código. Esta prueba es solo el comienzo.

- 4.3.1.9 LABORATORIO: Números primos: ¿Cómo encontrarlos?

LABORATORIO

Tiempo Estimado

15-20 minutos

Nivel de Dificultad

Medio

Objetivos

- Familiarizar al estudiante con nociones y algoritmos clásicos.
- Mejorar las habilidades del estudiante para definir y emplear funciones.

Escenario

Un número natural es **primo** si es mayor que 1 y no tiene divisores más que 1 y si mismo.

¿Complicado? De ninguna manera. Por ejemplo, 8 no es un número primo, ya que puedes dividirlo entre 2 y 4 (no podemos usar divisores iguales a 1 y 8, ya que la definición lo prohíbe).

Por otra parte, 7 es un número primo, ya que no podemos encontrar ningún divisor para el.

Tu tarea es escribir una función que verifique si un número es primo o no.

La función:

- Se llama `is_prime`.
- Toma un argumento (el valor a verificar).
- Devuelve `True` si el argumento es un número primo, y `False` de lo contrario.

Sugerencia: intenta dividir el argumento por todos los valores posteriores (comenzando desde 2) y verifica el resto: si es cero, tu número no puede ser un número primo; analiza cuidadosamente cuándo deberías detener el proceso.

Si necesitas conocer la raíz cuadrada de cualquier valor, puedes utilizar el operador `**`.
Recuerda: la raíz cuadrada de x es lo mismo que $x^{0.5}$.

Complementa el código en el editor.

Ejecuta tu código y verifica si tu salida es la misma que la nuestra.

Datos de prueba

Salida esperada:

```
2 3 5 7 11 13 17 19
```

- 4.3.1.10 LAB: Convirtiendo el consumo de combustible

•

LABORATORIO

Tiempo Estimado

10-15 minutos

Nivel de Dificultad

Fácil

Objetivos

- Mejorar las habilidades del estudiante para definir, utilizar y probar funciones.

Escenario

El consumo de combustible de un automóvil se puede expresar de muchas maneras diferentes. Por ejemplo, en Europa, se muestra como la cantidad de combustible consumido por cada 100 kilómetros.

En los EE. UU., se muestra como la cantidad de millas recorridas por un automóvil con un galón de combustible.

Tu tarea es escribir un par de funciones que conviertan l/100km a mpg (milas por galón), y viceversa.

Las funciones:

- Se llaman `liters_100km_to_miles_gallon` y `miles_gallon_to_liters_100km` respectivamente.
- Toman un argumento (el valor correspondiente a sus nombres).

Complementa el código en el editor.

Ejecuta tu código y verifica si tu salida es la misma que la nuestra.

Aquí hay información para ayudarte:

- 1 milla = 1609.344 metros.
- 1 galón = 3.785411784 litros.

Salida esperada:

60.31143162393162

31.36194444444444

23.52145833333333

3.9007393587617467

7.490910297239916

10.009131205673757