

- 2.4.1.7 LABORATORIO: Variables

-

LABORATORIO

Tiempo Estimado

10 minutos

Nivel de Dificultad

Fácil

Objetivos

- Familiarizarse con el concepto de almacenar y trabajar con diferentes tipos de datos en Python.
- Experimentar con el código en Python.

Escenario

A continuación una historia:

Érase una vez en la Tierra de las Manzanas, Juan tenía tres manzanas, María tenía cinco manzanas, y Adán tenía seis manzanas. Todos eran muy felices y vivieron por muchísimo tiempo. Fin de la Historia.

Tu tarea es:

- Crear las variables: `juan`, `maria`, y `adan`.
- Asignar valores a las variables. El valor debe de ser igual al número de manzanas que cada quien tenía.
- Una vez almacenados los números en las variables, imprimir las variables en una línea, y separar cada una de ellas con una coma.
- Después se debe crear una nueva variable llamada `total_manzanas` y se debe igualar a la suma de las tres variables anteriores.
- Imprime el valor almacenado en `total_manzanas` en la consola.
- **Experimenta con tu código:** crea nuevas variables, asigna diferentes valores a ellas, y realiza varias operaciones aritméticas con ellas (por ejemplo, `+`, `-`, `*`, `/`, `//`, etc.). Intenta poner una cadena con un entero juntos en la misma línea, por ejemplo, `"Número Total de Manzanas:"` y `total_manzanas`.

- 2.4.1.9 LABORATORIO: Variables, un sencillo convertidor

•

LABORATORIO

Tiempo Estimado

10 minutos

Nivel de Dificultad

Fácil

Objetivos

- Familiarizarse con el concepto de variables y trabajar con ellas.
- Realizar operaciones básicas y conversiones.
- Experimentar con el código de Python.

Escenario

Millas y kilómetros son unidades de longitud o distancia.

Teniendo en mente que 1 milla equivale aproximadamente a 1.61 kilómetros, complementa el programa en el editor para que convierta de:

- Millas a kilómetros.
- Kilómetros a millas.

No se debe cambiar el código existente. Escribe tu código en los lugares indicados con `###`. Prueba tu programa con los datos que han sido provistos en el código fuente.

Pon mucha atención a lo que esta ocurriendo dentro de la función `print()`. Analiza como es que se proveen múltiples argumentos para la función, y como es que se muestra el resultado.

Nota que algunos de los argumentos dentro de la función `print()` son cadenas (por ejemplo `"millas son"`), y otros son variables (por ejemplo `miles`).

TIP

Hay una cosa interesante más que esta ocurriendo. ¿Puedes ver otra función dentro de la función `print()`? Es la función `round()`. Su trabajo es redondear la salida del resultado al número de decimales especificados en el paréntesis, y regresar un valor flotante (dentro de la función `round()` se puede encontrar el nombre de la variable, el nombre, una coma, y el número de decimales que se desean mostrar). Se hablará más de esta función muy pronto, no te preocupes si no todo queda muy claro. Solo se quiere impulsar tu curiosidad.

Después de completar el laboratorio, abre Sandbox, y experimenta más. Intenta escribir diferentes convertidores, por ejemplo, un convertidor de USD a EUR, un convertidor de temperatura, etc. ¡Deja que tu imaginación vuele! Intenta mostrar los resultados combinando cadenas y variables. Intenta utilizar y experimentar con la función `round()` para redondear tus resultados a uno, dos o tres decimales. Revisa que es lo que sucede si no se provee un dígito al redondear. Recuerda probar tus programas.

Experimenta, saca tus propias conclusiones, y aprende. Sé curioso.

Resultado Esperado

```
7.38 millas son 11.88 kilómetros
12.25 kilómetros son 7.61 millas
```

- 2.4.1.10 LABORATORIO: Operadores y expresiones

•

LABORATORIO

Tiempo Estimado

10-15 minutos

Nivel de Dificultad

Fácil

Objetivos

- Familiarizarse con los conceptos de números, operadores y operaciones aritméticas en Python.
- Realizar cálculos básicos.

Escenario

Observa el código en el editor: lee un valor `flotante`, lo coloca en una variable llamada `x`, e imprime el valor de la variable llamada `y`. Tu tarea es completar el código para evaluar la siguiente expresión:

$$3x^3 - 2x^2 + 3x - 1$$

El resultado debe ser asignado a `y`.

Recuerda que la notación algebraica clásica muy seguido omite el operador de multiplicación, aquí se debe de incluir de manera explícita. Nota como se cambia el tipo de dato para asegurarnos de que `x` es del tipo `flotante`.

Mantén tu código limpio y legible, y pruébalo utilizando los datos que han sido proporcionados. No te desanimes por no lograrlo en el primer intento. Se persistente y curioso.

Datos de Prueba

Entrada de Muestra

x = 0
x = 1
x = -1

Salida Esperada

y = -1.0
y = 3.0
y = -9.0