

LAB-003-Exemplar_Practice writing Python code

September 20, 2024

1 Ejemplo: Práctica de escritura de código Python

1.1 Introducción

Python es un lenguaje de programación que ayuda a automatizar instrucciones para la computadora en una variedad de contextos, incluidos los de seguridad. Escribir código en Python es una habilidad valiosa que ayuda a los analistas de seguridad a prosperar en el aspecto técnico de su trabajo.

En este laboratorio, practicarás la escritura de tu primer código Python mientras aprendes sobre un entorno de cuaderno (o notebook). La práctica que realizas a lo largo de los laboratorios te ayudará a aplicar las habilidades de programación con Python a tu trabajo como analista de seguridad. Para aprovechar los laboratorios al máximo, asegúrate de no limitarte a escribir en las celdas que se te pide que completes, sino también de analizar todas las celdas a fondo.

Consejos para completar este laboratorio

Mientras recorres este laboratorio, ten en cuenta los siguientes consejos:

— **### TU CÓDIGO AQUÍ ###** indica dónde debes escribir el código. Asegúrate de reemplazarlo con tu propio código antes de ejecutar la celda de código. — Siéntete libre de abrir las pistas para obtener información adicional a medida que trabajas en cada tarea. — Para ingresar tu respuesta a una pregunta, haz doble clic en la celda de markdown para editar. Asegúrate de reemplazar “[Haz doble clic para ingresar aquí tus respuestas.]” con tu propia respuesta. — Puedes guardar tu trabajo manualmente con un clic en Archivo y luego en Guardar en la barra de menú en la parte superior del cuaderno. — Puedes descargar tu trabajo localmente con un clic en Archivo y luego en Descargar, luego puedes especificar el formato de archivo que prefieras en la barra de menú en la parte superior del cuaderno.

1.2 Situación hipotética

Como analista de seguridad, a menudo usarás cuadernos y entornos de cuadernos para escribir y ejecutar código. Este laboratorio te ayudará a familiarizarte con el trabajo en un entorno de cuaderno, escribir comentarios sobre el código en Python y mostrar cadenas con la función `print()`.

En este laboratorio, completarás una serie de tareas que implican observar y ejecutar algunas celdas de texto y código escritas previamente, así como rellenar celdas con tu propio texto, código Python y comentarios sobre el código.

1.3 Tarea 1

El entorno de laboratorio en el que estás trabajando es un entorno de codificación basado en cuadernos. Los cuadernos, como este, constan de dos tipos de celdas: (1) celdas de texto, también conocidas como celdas de markdown, y (2) celdas de código.

Las celdas de markdown te permiten escribir texto sin formato y formatearlo en lenguaje markdown. El lenguaje markdown se utiliza para formatear texto plano en editores de texto y editores de código. Por ejemplo, puedes usar markdown para crear encabezados, formatear palabras en negrita o cursiva, formatear texto como código, agregar hipervínculos y más.

Para esta tarea, escribe algo en la siguiente celda markdown. Asegúrate de reemplazar el “[Haz doble clic para editar esta celda markdown y escribe algo aquí.]” con tu propia respuesta.

Esta es una celda markdown y puedes ingresar texto aquí.

Pista 1

Haz doble clic en la celda markdown y reemplaza la sentencia del marcador de posición con la tuya. Puedes escribir la sentencia que tú quieras.

1.4 Tarea 2

En los cuadernos de Python, las celdas de código te permiten escribir comentarios sobre el código y código en Python.

Para ejecutar una celda de código, primero debes colocar el cursor sobre ella. A continuación, puedes hacer clic en el icono de reproducción o presionar las teclas *Mayús* e *Intro* (o, en algunos teclados, las teclas *Mayús* y *Retorno*).

Para esta tarea, ejecuta la siguiente celda de código tal como está y observa el resultado.

```
[1]: # Esta celda muestra ";Hola, mundo!"  
  
print(";Hola, mundo!")
```

¡Hola, mundo!

Pista 1

Una vez que haces clic en la celda de código, puedes ejecutar el código haciendo clic en el ícono de reproducción triangular o presionando las teclas *Mayús* e *Intro* (o, en algunos teclados, las teclas *Mayús* y *Retorno*).

Pregunta 1 ¿Qué observas acerca del resultado después de ejecutar la celda de código?

El resultado muestra el enunciado ¡Hola, mundo! en pantalla.

1.5 Tarea 3

Escribir comentarios sobre el código es una forma de documentar la intención detrás del código. Es un estándar que los analistas suelen usar en su flujo de trabajo. Escribir comentarios que acompañen el código también te permite hacer un seguimiento de las decisiones técnicas que tomaste en tu proyecto. Esto hace que sea más fácil para ti y tu equipo leer y revisar tu código para comprender qué hace y por qué adoptaste ciertos enfoques.

Para esta tarea, ejecuta la siguiente celda de código tal como está y observa el resultado.

```
[2]: # En Python, no se muestran los comentarios
     # Esta celda de código contiene solo comentarios
```

Pista 1

Una vez que haces clic en la celda de código, puedes ejecutar el código haciendo clic en el ícono de reproducción triangular o presionando las teclas *Mayús* e *Intro* (o, en algunos teclados, las teclas *Mayús* y *Retorno*).

Pregunta 2 ¿Qué observas acerca del resultado después de ejecutar la celda anterior?

No hay resultado. Esto se debe a que la celda de código anterior consta de dos comentarios sobre el código. Cada comentario comienza con un símbolo de almohadilla (#). Al ejecutar código, Python ignora los comentarios. Los comentarios de código no son interpretados por las computadoras, sino por los analistas.

1.6 Tarea 4

Para escribir en una celda de código, primero debes hacer clic en ella. Luego, puedes escribir comentarios y código dentro de la celda.

Para esta tarea, agrega un comentario al comienzo de la siguiente celda de código describiendo qué hace el código. Escribe el comentario para decir `# Esta celda muestra "Estoy usando Python."`. Asegúrate de reemplazar el fragmento `# TU COMENTARIO AQUÍ` con tu propio comentario antes de ejecutar la siguiente celda.

```
[3]: # Esta celda muestra "Estoy usando Python."
     print("Estoy usando Python.")
```

Estoy usando Python.

Pista 1

Una vez que hagas clic en la celda de código, reemplaza el comentario del marcador de posición con tu comentario. Recuerda que los comentarios en Python comienzan con el símbolo de almohadilla (#).

Pregunta 3 ¿Qué observas acerca del resultado después de ejecutar la celda anterior?

El resultado es `Estoy usando Python`. Ten en cuenta que las comillas que encierran una cadena no aparecen en el resultado al mostrar la cadena.

`##` Tarea 5

En Python, `print()` te ayuda a mostrar información en la pantalla.

Para esta tarea, usa `print()` para mostrar el mensaje `"Soy un analista de seguridad"` colocando ese mensaje entre los paréntesis. Asegúrate de reemplazar el fragmento `### TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda de código.

```
[4]: # Esta celda muestra "Soy analista de seguridad."  
  
print("Soy analista de seguridad.")
```

`Soy analista de seguridad.`

Pista 1

Una vez que hagas clic en la celda de código, reemplaza el fragmento `### TU CÓDIGO AQUÍ ###` con `"Soy un analista de seguridad."`.

Pregunta 4 ¿Qué observas acerca del resultado después de ejecutar la celda anterior?

El resultado es `Soy analista de seguridad`. Ten en cuenta que las comillas que encierran una cadena no aparecen en el resultado al mostrar la cadena.

1.7 Tarea 6

Para esta tarea, escribe una sentencia `print()` para mostrar la cadena `"¡Python es útil para la seguridad!"`. Asegúrate de reemplazar el fragmento `### TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda de código.

```
[5]: # Esta celda muestra "¡Python es útil para la seguridad!"  
  
print("¡Python es útil para la seguridad!")
```

`¡Python es útil para la seguridad!`

Pista 1

Haz clic en la celda de código y utiliza `print()` para mostrar el mensaje.

Pista 2

Coloca el mensaje entre los paréntesis de `print()`.

Pregunta 5 ¿Qué observas acerca del resultado después de ejecutar la celda anterior?

El resultado es `¡Python es útil para la seguridad!` Ten en cuenta que las comillas que encierran una cadena no aparecen en el resultado al mostrar la cadena.

1.8 Tarea 7

Para tu última tarea, combinarás todas las sentencias `print()` que hayas encontrado y escrito en este laboratorio hasta ahora en una celda de código.

Completa el siguiente código con los mensajes restantes. Asegúrate de reemplazar cada `### TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda.

```
[6]: # Esta celda muestra todas las sentencias escritas hasta ahora

print("¡Hola, mundo!")
print("Estoy usando Python.")
print("Soy analista de seguridad.")
print("¡Python es útil para la seguridad!")
```

```
¡Hola, mundo!
Estoy usando Python.
Soy analista de seguridad.
¡Python es útil para la seguridad!
```

Pista 1

Primero, haz clic en la celda de código.

Para la cuarta sentencia de `print()`, coloca la cadena `"Soy un analista de seguridad."` dentro de los paréntesis de `print()`.

Para la quinta sentencia de `print()`, coloca la cadena `"Python es útil para la seguridad!"` dentro de los paréntesis de `print()`.

Pregunta 6 ¿Qué observas acerca del resultado después de ejecutar la celda anterior?

Cada línea del resultado resultó de una sentencia `print()` en el código. `¡Hola, mundo!` aparece en la primera línea del resultado, `Estoy usando Python.` aparece en la siguiente línea del resultado, `Soy analista de seguridad.` aparece a continuación y, por último, `¡Python es útil para la seguridad!` aparece al final. Ten en cuenta que Python interpreta y ejecuta el código en orden, de arriba a abajo y de izquierda a derecha.

1.9 Conclusión

¿Qué conclusiones clave obtuviste de este laboratorio?

Es útil usar comentarios sobre el código para documentar las decisiones que tomas a medida que programas. Las computadoras ignoran los comentarios sobre el código. Solo tú y tu equipo podrán leerlos para comprender lo que hace el código. — Puedes escribir comentarios en Python con el

símbolo de almohadilla o hash (#). — Puedes usar `print()` en Python para mostrar información en pantalla. — Al utilizar `print()` para mostrar una cadena, las comillas que encierran la cadena no aparecen en el resultado en pantalla.