

LAB-004-Exemplar_Assign Python variables

September 20, 2024

1 Ejemplo: Asignación de variables de Python

1.1 Introducción

Las variables ayudan a los analistas de seguridad a realizar un seguimiento de una variedad de información relacionada con la seguridad. Por ejemplo, es posible que un analista tenga que crear variables de Python para los usuarios a los que se les permite iniciar sesión, cuántos intentos de inicio de sesión tienen en total y la cantidad actual de veces que intentaron iniciar sesión.

En este laboratorio, practicarás cómo asignar valores a variables y determinar sus tipos de datos.

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

Eres un analista de seguridad y tienes la responsabilidad de escribir código que automatice el análisis de los intentos de inicio de sesión realizados en un dispositivo específico. Como primer paso, deberás crear variables para realizar un seguimiento de la información relevante para el proceso de inicio de sesión. Esta información incluye el ID del dispositivo, la lista de nombres de usuario aprobados, los intentos de inicio de sesión máximos permitidos por usuario, los intentos de inicio de sesión actuales realizados por un usuario y el estado de inicio de sesión.

A lo largo de este laboratorio, asignarás estas variables y verificarás los tipos de datos de estas.

1.3 Tarea 1

Imagina que, en tu trabajo como analista, hay un dispositivo al que solo pueden acceder los usuarios especificados en una lista de permisos y que el ID del dispositivo es "72e08x0".

En la siguiente celda de código, asigna este valor a una variable llamada `id_dispositivo`. Luego, muestra el contenido de la variable y observa el resultado.

Asegúrate de reemplazar cada `## TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda.

```
[1]: # Asigna a la variable `id_dispositivo` el ID del dispositivo al que solo los
      ↪ usuarios especificados pueden acceder

      id_dispositivo = "72e08x0"

      # Muestra `id_dispositivo`

      print(id_dispositivo)
```

72e08x0

Pista 1

Usa comillas alrededor del valor que asignas a la variable `id_dispositivo`.

Pista 2

Asigna el valor "72e08x0" a la variable `id_dispositivo`. Para hacerlo, indica el valor a la derecha del operador `=`.

Pista 3

Para mostrar el contenido de `id_dispositivo`, llama a la función `print()` e indica `id_dispositivo` como argumento.

1.4 Tarea 2

Ahora que la variable `id_dispositivo` está definida, puedes devolver el tipo de dato al que se refiere.

En esta tarea, usa una función de Python para encontrar el tipo de dato de la variable `id_dispositivo`. Luego, almacena el tipo de dato en otra variable llamada `tipo_id_dispositivo` y muéstralo para observar el resultado.

Asegúrate de reemplazar cada `## TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda.

```
[2]: # Asigna a la variable `id_dispositivo` el ID del dispositivo al que solo los
      ↪ usuarios especificados pueden acceder

      id_dispositivo = "72e08x0"
```

```
# Asigna a `tipo_id_dispositivo` el tipo de dato de `id_dispositivo`

tipo_id_dispositivo = type(id_dispositivo)

# Muestra `tipo_id_dispositivo`

print(tipo_id_dispositivo)
```

<class 'str'>

Pista 1

La función `type()` te permite obtener el tipo de dato de un valor determinado.

Pista 2

Utiliza `type(id_dispositivo)` para obtener el tipo de dato de la variable `id_dispositivo`.

Pista 3

Para mostrar `tipo_id_dispositivo`, coloca el nombre de esta variable dentro de la función `print()`.

Pregunta 1 A partir del resultado anterior, ¿qué observas acerca del tipo de dato de `id_dispositivo`?

El resultado anterior muestra que el tipo de dato de `id_dispositivo` es `str`. Eso significa que `id_dispositivo` almacena un valor de cadena.

1.5 Tarea 3

A medida que continúas con tu trabajo, te dan una lista de nombres de usuario de los usuarios que tienen permitido acceder al dispositivo. Los nombres de usuario con este acceso son "madebowa", "jnguyen", "tbecker", "nhersh" y "redwards".

Para esta tarea, crea una variable llamada `lista_usuarios`. Luego, crea una lista con los nombres de usuario aprobados y asígnala a esta variable.

Asegúrate de reemplazar cada `## TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda.

```
[3]: # Asigna `lista_usuarios` a la lista de nombres de usuario a los que se les  
      ↪permite acceder al dispositivo

lista_usuarios = ["madebowa", "jnguyen", "tbecker", "nhersh", "redwards"]

# Muestra `lista_usuarios`

print(lista_usuarios)
```

```
['madebowa', 'jnguyen', 'tbecker', 'nhersh', 'redwards']
```

Pista 1

Para crear una lista en Python, usa corchetes. Dentro de los corchetes, escribe los elementos de la lista, separados con comas.

Pista 2

Para asignar un valor a una variable en Python, coloca el nombre de la variable a la izquierda del operador = y el valor a la derecha del operador =.

En esta tarea, asegúrate de colocar `lista_usuarios` a la izquierda del operador =.

Pista 3

Para mostrar el valor de una variable, llama a la función `print()` e indica el nombre de la variable.

En esta tarea, asegúrate de llamar a la función `print()` e indicar `lista_usuarios`.

1.6 Tarea 4

Para esta tarea, busca el tipo de dato de `lista_usuarios`. Almacena el tipo en una variable llamada `tipo_lista_usuarios` y muéstralo para observar el resultado.

Asegúrate de reemplazar cada `## TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda.

```
[4]: # Asigna `lista_usuarios` a la lista de nombres de usuario a los que se les
      ↳ permite acceder al dispositivo
```

```
lista_usuarios = ["madebowa", "jnguyen", "tbecker", "nhersh", "redwards"]
```

```
# Asigna `tipo_lista_usuarios` al tipo de dato de `lista_usuarios`
```

```
tipo_lista_usuarios = type(lista_usuarios)
```

```
# Muestra `tipo_lista_usuarios`
```

```
print(tipo_lista_usuarios)
```

```
<class 'list'>
```

Pista 1

La función `type()` te permite obtener el tipo de dato de un valor determinado.

Pista 2

Para obtener el tipo de dato de `lista_usuarios`, llama a la función `type()` e indica `lista_usuarios`.

Asigna `lista_usuarios` a esa llamada a la función.

Pista 3

Para mostrar `tipo_lista_usuarios`, llama a la función `print()` y pásale `tipo_lista_usuarios` como argumento.

Pregunta 2 A partir del resultado anterior, ¿qué observas acerca del tipo de dato de `lista_usuarios`?

El resultado anterior muestra que el tipo de dato de `lista_usuarios` es `list`. Eso significa que `lista_usuarios` almacena una lista.

Tarea 5 Ahora, supongamos que te avisan de que la lista anterior no está actualizada y de que hay otro empleado que ahora tiene acceso al dispositivo. Te dan la siguiente lista actualizada de nombres de usuario con acceso, incluido el empleado nuevo: "madebowa", "jnguyen", "tbecker", "nhersh", "redwards" y "lpope".

Para esta tarea, reasigna la lista nueva a la variable `lista_usuarios`. Ejecuta el código para mostrar la lista antes y después de la actualización para observar la diferencia.

Asegúrate de reemplazar cada `## TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda.

```
[5]: # Asigna `lista_usuarios` a la lista de nombres de usuario a los que se les
    ↪permite acceder al dispositivo

lista_usuarios = ["madebowa", "jnguyen", "tbecker", "nhersh", "redwards"]

# Muestra `lista_usuarios`

print(lista_usuarios)

# Asigna a `lista_usuarios` la lista actualizada de nombres de usuario a los
    ↪que se les permite acceder al dispositivo

lista_usuarios = ["madebowa", "jnguyen", "tbecker", "nhersh", "redwards",
    ↪"lpope"]

# Muestra `lista_usuarios`

print(lista_usuarios)
```

```
['madebowa', 'jnguyen', 'tbecker', 'nhersh', 'redwards']
['madebowa', 'jnguyen', 'tbecker', 'nhersh', 'redwards', 'lpope']
```

Pista 1

Al reasignar una variable a un nuevo valor en Python, coloca el nombre de la variable a la izquierda del operador `=`, tal como lo harías al asignar la variable por primera vez.

Pista 2

Para reasignar `lista_usuarios` a la lista actualizada, coloca `lista_usuarios` a la izquierda del operador `=`.

Pregunta 3 A partir del resultado anterior, ¿qué observas acerca del contenido de `lista_usuarios`?

El resultado anterior muestra que el contenido de `lista_usuarios` se actualizó después de que se reasignó la lista nueva a la variable.

La primera llamada `print()` da como resultado el contenido original de la lista. La segunda llamada `print()` da como resultado el contenido actualizado, que incluye el nombre de usuario recién agregado, "lpope".

1.7 Tarea 6

Para esta tarea, define una variable llamada `max_inicios_sesion` que represente el número máximo de intentos de inicio de sesión permitidos por usuario. Almacena el valor 3 en esta variable. Luego, almacena el tipo de dato en otra variable llamada `tipo_max_inicios_sesion` y muéstralo para observar el resultado.

Asegúrate de reemplazar cada `## TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda.

```
[6]: # Asigna a `max_inicios_sesion` el valor 3

max_inicios_sesion = 3

# Asigna a `tipo_max_inicios_sesion` el tipo de dato de `max_inicios_sesion`

tipo_max_inicios_sesion = type(max_inicios_sesion)

# Muestra `tipo_max_inicios_sesion`

print(tipo_max_inicios_sesion)
```

```
<class 'int'>
```

Pista 1

Al asignar un valor a una variable en Python, utiliza el operador `=`. Coloca el nombre de la variable a la izquierda del operador `=` y el valor a la derecha del operador `=`.

Pista 2

Para asignar 3 a `max_inicios_sesion`, coloca `max_inicios_sesion` a la izquierda del operador `=` y 3 a la derecha del operador `=`.

Para asignar un valor a `tipo_max_inicios_sesion`, coloca `tipo_max_inicios_sesion` a la izquierda del operador `=` antes de la llamada a la función `type()`.

Pista 3

En Python, puedes utilizar la función `print()` para mostrar el valor de una variable.

Para mostrar `tipo_max_inicios_sesion`, llama a `print()` y pásale `tipo_max_inicios_sesion` como argumento.

Pregunta 4 A partir del resultado anterior, ¿qué observas acerca del tipo de dato de `max_inicios_sesion`?

El resultado anterior muestra que el tipo de dato de `max_inicios_sesion` es `int`. Eso significa que `max_inicios_sesion` almacena un valor entero.

1.8 Tarea 7

En esta tarea, define una variable llamada `intentos_inicio_sesion` que represente el número actual de intentos de inicio de sesión realizados por un usuario. Almacena el valor 2 en esta variable. Luego, almacena el tipo de dato `intentos_inicio_sesion` en una variable llamada `tipo_intentos_inicio_sesion` y, a continuación, imprímelo para observar el resultado.

Asegúrate de reemplazar cada `## TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda.

```
[2]: # Asigna `intentos_inicio_sesion` al valor 2

intentos_inicio_sesion = 2

# Asigna a `tipo_intentos_inicio_sesion` el tipo de dato de
↳ `intentos_inicio_sesion`

tipo_intentos_inicio_sesion = type(intentos_inicio_sesion)

# Muestra `tipo_intentos_inicio_sesion`

print(tipo_intentos_inicio_sesion)
```

```
<class 'int'>
```

Pista 1

Al asignar un valor a una variable en Python, utiliza el operador `=`. Coloca el nombre de la variable a la izquierda del operador `=` y el valor a la derecha del operador `=`.

Pista 2

Para asignar 2 a `intentos_inicio_sesion`, coloca `intentos_inicio_sesion` a la izquierda del operador `=` y 2 a la derecha del operador `=`.

Para asignar un valor a `tipo_intentos_inicio_sesion`, coloca `tipo_intentos_inicio_sesion` a la izquierda del operador `=` y llama a la función `type()` a la derecha del operador `=`.

Al llamar a `type()`, asegúrate de indicar `intentos_inicio_sesion`.

Pista 3

En Python, puedes utilizar la función `print()` para mostrar el valor de una variable.

Para mostrar `tipo_intentos_inicio_sesion`, llama a la función `print()` e indica `tipo_intentos_inicio_sesion`.

Pregunta 5 A partir del resultado anterior, ¿qué observas acerca del tipo de dato de `tipo_intentos_inicio_sesion`?

El resultado anterior muestra que el tipo de dato de `tipo_intentos_inicio_sesion` es `int`. Eso significa que `tipo_intentos_inicio_sesion` almacena un valor entero.

1.9 Tarea 8

En esta tarea, determinarás el valor booleano que representa si el número actual de intentos de inicio de sesión que realizó un usuario es menor o igual al número máximo de intentos de inicio de sesión permitidos.

Asegúrate de reemplazar cada `## TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda.

```
[8]: # Asigna a `max_inicios_sesion` el valor 3

max_inicios_sesion = 3

# Asigna `intentos_inicio_sesion` al valor 2

intentos_inicio_sesion = 2

# Determina si el número actual de intentos de inicio de sesión que realizó un
  ↳ usuario es menor o igual al número máximo de intentos de inicio de sesión
  ↳ permitidos,
# y muestra el valor booleano resultante

print(intentos_inicio_sesion <= max_inicios_sesion)
```

Verdadero

Pista 1

En Python, puedes usar el operador de comparación `<=` para determinar si un valor es menor o igual que otro valor.

Pista 2

Para determinar si el número actual de intentos de inicio de sesión que realizó un usuario es menor o igual al número máximo de intentos de inicio de sesión permitidos, usa el operador `<=`. Coloca `intentos_inicio_sesion` a la izquierda del operador `<=` y `max_inicios_sesion` a la derecha del operador `<=`.

Para asegurarte de que se muestre el valor booleano resultante, escribe este código dentro de los paréntesis al llamar a la función `print()`.

Pregunta 6 ¿Cuál es el resultado? ¿Qué significa esto?

El resultado anterior es **Verdadero**, lo que indica que `intentos_inicio_sesion` es menor o igual que `max_inicios_sesion`. En otras palabras, el número actual de intentos que el usuario realizó para iniciar sesión aún no excedió el número máximo de intentos permitidos.

1.10 Tarea 9

Este código continúa verificando el valor booleano de si `max_inicios_sesion` es menor o igual que `intentos_inicio_sesion`. En esta tarea, reasigna otros valores a `intentos_inicio_sesion`. Observa cómo cambia el resultado.

Asegúrate de reemplazar cada **## TU CÓDIGO AQUÍ ###** con tu propio código antes de ejecutar la siguiente celda.

```
[9]: # Asigna a `max_inicios_sesion` el valor 3

max_inicios_sesion = 3

# Asigna `intentos_inicio_sesion` a un valor específico

intentos_inicio_sesion = 4

# Determina si el número actual de intentos de inicio de sesión que realizó un
→usuario es menor o igual al número máximo de intentos de inicio de sesión
→permitidos,
# y muestra el valor booleano resultante

print(intentos_inicio_sesion <= max_inicios_sesion)
```

Falso

Pista 1

Para asignar un valor específico a `intentos_inicio_sesion`, asegúrate de escribir el valor a la derecha del operador `=`.

Pregunta 7 A partir de los diferentes valores que asignaste a `intentos_inicio_sesion`, ¿qué observaste acerca del resultado?

El valor booleano en el resultado cambia según el valor asignado a `intentos_inicio_sesion`. Por ejemplo, cuando `intentos_inicio_sesion` se asigna a 4, el resultado es **Falso**, lo que indica que `intentos_inicio_sesion` no es menor o igual que `max_inicios_sesion`. En otras palabras, el número actual de intentos que el usuario realizó para iniciar sesión excedió el número máximo de intentos permitidos.

1.11 Tarea 10

Por último, también puedes asignar un valor booleano de **Verdadero** o **Falso** a una variable.

En esta tarea, crearás una variable llamada `estado_inicio_sesion`, que es un valor booleano que representa si un usuario está conectado. Asigna `Falso` a esta variable, almacena el tipo de dato en una variable llamada `tipo_estado_inicio_sesion` y muéstrala.

Asegúrate de reemplazar cada `## TU CÓDIGO AQUÍ ###` con tu propio código antes de ejecutar la siguiente celda.

```
[10]: # Asigna a `estado_inicio_sesion` el valor booleano `Falso`

estado_inicio_sesion = Falso

# Asigna a `tipo_estado_inicio_sesion` el tipo de dato de `estado_inicio_sesion`

tipo_estado_inicio_sesion = type(estado_inicio_sesion)

# Muestra `tipo_estado_inicio_sesion`

print(tipo_estado_inicio_sesion)
```

```
<class 'bool'>
```

Pista 1

Para asignar el valor booleano `Falso` a la variable `estado_inicio_sesion`, asegúrate de escribir `Falso` a la derecha del operador `=`.

Pista 2

Ten en cuenta que los valores booleanos no deben tener comillas alrededor en el código.

Pregunta 8 A partir del resultado anterior, ¿qué observas acerca del tipo de dato de `estado_inicio_sesion`?

El resultado anterior muestra que el tipo de dato de `estado_inicio_sesion` es `bool`. Eso significa que `estado_inicio_sesion` almacena un valor booleano.

1.12 Conclusión

¿Qué conclusiones clave obtuviste de este laboratorio?

— Hay muchos operadores útiles en Python que te ayudan a trabajar con variables. — El operador de asignación `=` te permite asignar o reasignar un valor específico a una variable. — El operador de comparación `<=` te permite comparar el valor de una variable con el valor de otra. — La función `type()` en Python te ayuda a determinar el tipo de dato de un objeto. — Si le indicas una variable a `type()`, mostrará el tipo de dato del valor almacenado en la variable. — La función `print()` en Python te permite mostrar información. — Puede tomar un valor directamente, una variable que almacena un valor o una comparación entre variables que se evalúa a un valor booleano.