

# Activity: Assign Python variables

## Introduction

Variables help security analysts to keep track of a variety of security-related information. For example, analysts may need to create Python variables for the users who are allowed to log in, the number of login attempts that they're permitted, and the current number of attempts that a user has made.

In this lab, you'll practice assigning values to variables and determining their data types.

## Tips for completing this lab

- 
- 
- 
- 
- 

## Scenario

You are a security analyst who is responsible for writing code that will automate analysis of login attempts made to a specific device. As the first step, you'll need to create variables to keep track of information relevant to the login process. This information includes the device ID, list of approved usernames, maximum login attempts allowed per user, current login attempts made by a user, and login status.

Throughout this lab, you'll assign these variables and check the data types of the variables.

## Task 1

In your work as an analyst, imagine there is a device only users specified on an allow list can access, and its device ID is "72e08x0".

In the following code cell, assign this value to a variable named `device_id`. Then, display the contents of the variable and observe the output.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

```
In [ ]:
# Assign the `device_id` variable to the device ID that only specified users
can access

device_id = ### YOUR CODE HERE ###

# Display `device_id`
```

```
print(### YOUR CODE HERE ###)
```

Hint 1

Hint 2

Hint 3

## Task 2

Now that the variable `device_id` is defined, you can return its data type.

In this task, use a Python function to find the data type of the variable `device_id`. Store the data type in another variable called `device_id_type`. Then, display `device_id_type` to examine the output.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

```
In [ ]:  
# Assign the `device_id` variable to the device ID that only specified users  
# can access  
  
device_id = "72e08x0"  
  
# Assign `device_id_type` to the data type of `device_id`  
  
device_id_type = ### YOUR CODE HERE ###  
  
# Display `device_id_type`  
  
print(### YOUR CODE HERE ###)
```

Hint 1

Hint 2

Hint 3

Question 1

Based on the output above, what do you observe about the data type of `device_id`?

[Double-click to enter your responses here.]

## Task 3

As you continue your work, you're provided a list of usernames of users who are allowed to access the device. The usernames with this access are "madebowa", "jnguyen", "tbecker", "nhersh", and "redwards".

In this task, create a variable called `username_list`. Assign a list with the approved usernames to this variable. Then, display the value of the `username_list` variable.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [ ]:

```
# Assign `username_list` to the list of usernames who are allowed to access
the device

username_list = ### YOUR CODE HERE ###

# Display `username_list`

print(### YOUR CODE HERE ###)
```

**Hint 1**

**Hint 2**

**Hint 3**

## Task 4

In this task, find the data type of the `username_list`. Store the type in a variable called `username_list_type`. Then, display `username_list_type` to examine the output.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [ ]:

```
# Assign `username_list` to the list of usernames who are allowed to access
the device

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

# Assign `username_list_type` to the data type of `username_list`

username_list_type = ### YOUR CODE HERE ###

# Display `username_list_type`

### YOUR CODE HERE ###
```

Hint 1

Hint 2

Hint 3

## Question 2

Based on the output above, what do you observe about the data type of `username_list`?

[Double-click to enter your responses here.]

## Task 5

Now, imagine that you've been informed that the previous list is not up-to-date and that there is another employee that now has access to the device. You're given the updated list of usernames with access, including the new employee, as follows: `"madebowa"`, `"jnguyen"`, `"tbecker"`, `"nhersh"`, `"redwards"`, and `"lpope"`.

In this task, reassign the variable `username_list` to the new list. Run the code to display the list before and after it's been updated to observe the difference.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [ ]:

```
# Assign `username_list` to the list of usernames who are allowed to access the device

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

# Display `username_list`

print(username_list)

# Assign `username_list` to the updated list of usernames who are allowed to access the device

### YOUR CODE HERE ### = ["madebowa", "jnguyen", "tbecker", "nhersh", "redwards", "lpope"]

# Display `username_list`

print(username_list)
```

Hint 1

## Hint 2

## Question 3

Based on the output above, what do you observe about the contents of `username_list`?

[Double-click to enter your responses here.]

## Task 6

In this task, define a variable called `max_logins` that represents the maximum number of login attempts allowed per user. Store the value `3` in this variable. Then, store its data type in another variable called `max_logins_type`. Display `max_logins_type` to examine the output.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [ ]:

```
# Assign `max_logins` to the value 3

### YOUR CODE HERE ###

# Assign `max_logins_type` to the data type of `max_logins`

### YOUR CODE HERE ### = type(max_logins)

# Display `max_logins_type`

### YOUR CODE HERE ###(max_logins_type)
```

## Hint 1

## Hint 2

## Hint 3

## Question 4

Based on the output above, what do you observe about the data type of `max_logins`?

[Double-click to enter your responses here.]

## Task 7

In this task, define a variable called `login_attempts` that represents the current number of login attempts made by a user. Store the value `2` in this variable. Then, store its data type in a variable called `login_attempts_type`. Display `login_attempts_type` to observe the output.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [ ]:

```
# Assign `login_attempts` to the value 2

### YOUR CODE HERE ###

# Assign `login_attempts_type` to the data type of `login_attempts`

### YOUR CODE HERE ###

# Display `login_attempts_type`

### YOUR CODE HERE ###
```

Hint 1

Hint 2

Hint 3

Question 5

Based on the output above, what do you observe about the data type of `login_attempts`?  
[Double-click to enter your responses here.]

## Task 8

In this task, you'll determine the Boolean value that represents whether the current number of login attempts a user has made is less than or equal to the maximum number of login attempts allowed.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [ ]:

```
# Assign `max_logins` to the value 3

max_logins = 3

# Assign `login_attempts` to the value 2

login_attempts = 2

# Determine whether the current number of login attempts a user has made is
less than or equal to the maximum number of login attempts allowed,
# and display the resulting Boolean value
```

```
print(### YOUR CODE HERE ### <= ### YOUR CODE HERE ###)
```

Hint 1

Hint 2

### Question 6

**What is the output? What does this mean?**

[Double-click to enter your responses here.]

## Task 9

This code continues to check for the Boolean value of whether `max_logins` is less than or equal to `login_attempts`. In this task, reassign other values to `login_attempts`. For example, you might choose a value that is higher than the maximum number of attempts allowed. Observe how the output changes.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [ ]:

```
# Assign `max_logins` to the value 3

max_logins = 3

# Assign `login_attempts` to a specific value

login_attempts = ### YOUR CODE HERE ###

# Determine whether the current number of login attempts a user has made is
# less than or equal to the maximum number of login attempts allowed,
# and display the resulting Boolean value

print(login_attempts <= max_logins)
```

Hint 1

### Question 7

**Based on the different values you assigned to `login_attempts`, what did you observe about the output?**

[Double-click to enter your responses here.]

## Task 10

Finally, you can also assign a Boolean value of `True` or `False` to a variable.

In this task, you'll create a variable called `login_status`, which is a Boolean that represents whether a user is logged in. Assign `False` to this variable and store its data type in a variable called `login_status_type` and display it.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [ ]:

```
# Assign `login_status` to the Boolean value False

login_status = ### YOUR CODE HERE ###

# Assign `login_status_type` to the data type of `login_status`

login_status_type = type(login_status)

# Display `login_status_type`

print(login_status_type)
```

Hint 1

Hint 2

Question 8

Based on the output above, what do you observe about the data type of `login_status`?

[Double-click to enter your responses here.]

## Conclusion

What are your key takeaways from this lab?

[Double-click to enter your responses here.]

# Exemplar: Assign Python variables

## Introduction

Variables help security analysts to keep track of a variety of security-related information. For example, analysts may need to create Python variables for the users who are allowed to log in, the number of login attempts that they're permitted, and the current number of attempts that a user has made.

In this lab, you'll practice assigning values to variables and determining their data types.



## Tips for completing this lab

- 
- 
- 
- 
- 

## Scenario

You are a security analyst who is responsible for writing code that will automate analysis of login attempts made to a specific device. As the first step, you'll need to create variables to keep track of information relevant to the login process. This information includes the device ID, list of approved usernames, maximum login attempts allowed per user, current login attempts made by a user, and login status.

Throughout this lab, you'll assign these variables and check the data types of the variables.

## Task 1

In your work as an analyst, imagine there is a device only users specified on an allow list can access, and its device ID is "72e08x0".

In the following code cell, assign this value to a variable named `device_id`. Then, display the contents of the variable and observe the output.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [1]:

```
# Assign the `device_id` variable to the device ID that only specified users  
can access
```

```
device_id = "72e08x0"
```

```
# Display `device_id`
```

```
print(device_id)
```

72e08x0

Hint 1

Hint 2

Hint 3

## Task 2

Now that the variable `device_id` is defined, you can return its data type.

In this task, use a Python function to find the data type of the variable `device_id`. Store the data type in another variable called `device_id_type`. Then, display `device_id_type` to examine the output.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [2]:

```
# Assign the `device_id` variable to the device ID that only specified users  
can access
```

```
device_id = "72e08x0"
```

```
# Assign `device_id_type` to the data type of `device_id`
```

```
device_id_type = type(device_id)
```

```
# Display `device_id_type`
```

```
print(device_id_type)
```

```
<class 'str'>
```

Hint 1

Hint 2

Hint 3

Question 1

**Based on the output above, what do you observe about the data type of `device_id`?**

The output above shows that the data type of `device_id` is `str`, which means that `device_id` stores a string value.

## Task 3

As you continue your work, you're provided a list of usernames of users who are allowed to access the device. The usernames with this access are `"madebowa"`, `"jnguyen"`, `"tbecker"`, `"nhersh"`, and `"redwards"`.

In this task, create a variable called `username_list`. Assign a list with the approved usernames to this variable. Then, display the value of the `username_list` variable.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [3]:

```
# Assign `username_list` to the list of usernames who are allowed to access the device
```

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

```
# Display `username_list`
```

```
print(username_list)
```

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

**Hint 1**

**Hint 2**

**Hint 3**

## Task 4

In this task, find the data type of the `username_list`. Store the type in a variable called `username_list_type`. Then, display `username_list_type` to examine the output.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [4]:

```
# Assign `username_list` to the list of usernames who are allowed to access the device
```

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

```
# Assign `username_list_type` to the data type of `username_list`
```

```
username_list_type = type(username_list)
```

```
# Display `username_list_type`
```

```
print(username_list_type)
```

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

Hint 1

Hint 2

Hint 3

## Question 2

Based on the output above, what do you observe about the data type of `username_list`?

The output above shows that the data type of `username_list` is `list`, which means that `username_list` stores a list.

## Task 5

Now, imagine that you've been informed that the previous list is not up-to-date and that there is another employee that now has access to the device. You're given the updated list of usernames with access, including the new employee, as follows: `"madebowa"`, `"jnguyen"`, `"tbecker"`, `"nhersh"`, `"redwards"`, and `"lpope"`.

In this task, reassign the variable `username_list` to the new list. Run the code to display the list before and after it's been updated to observe the difference.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [5]:

```
# Assign `username_list` to the list of usernames who are allowed to access
the device

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

# Display `username_list`

print(username_list)

# Assign `username_list` to the updated list of usernames who are allowed to
access the device

username_list = ["madebowa", "jnguyen", "tbecker", "nhersh", "redwards",
"lpope"]

# Display `username_list`

print(username_list)
```

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

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

**Hint 1**

**Hint 2**

**Question 3**

**Based on the output above, what do you observe about the contents of `username_list`?**

The output above shows that the contents of `username_list` were updated after the variable was reassigned to the new list.

The first `print()` call output the original contents of the list. The second `print()` call output the updated contents, which includes the newly added username, `"lpope"`.

## Task 6

In this task, define a variable called `max_logins` that represents the maximum number of login attempts allowed per user. Store the value `3` in this variable. Then, store its data type in another variable called `max_logins_type`. Display `max_logins_type` to examine the output.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [6]:

```
# Assign `max_logins` to the value 3

max_logins = 3

# Assign `max_logins_type` to the data type of `max_logins`

max_logins_type = type(max_logins)

# Display `max_login_type`

print(max_logins_type)
```

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

**Hint 1**

**Hint 2**

**Hint 3**

**Question 4**

Based on the output above, what do you observe about the data type of `max_logins`?

The output above shows that the data type of `max_logins` is `int`, which means that `max_logins` stores an integer value.

## Task 7

In this task, define a variable called `login_attempts` that represents the current number of login attempts made by a user. Store the value `2` in this variable. Then, store `login_attempts` data type in a variable called `login_attempts_type`. Display `login_attempts_type` to observe the output.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [2]:

```
# Assign `login_attempts` to the value 2

login_attempts = 2

# Assign `login_attempts_type` to the data type of `login_attempts`

login_attempts_type = type(login_attempts)

# Display `login_attempts_type`

print(login_attempts_type)
```

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

Hint 1

Hint 2

Hint 3

Question 5

Based on the output above, what do you observe about the data type of `login_attempts`?

The output above shows that the data type of `login_attempts` is `int`, which means that `login_attempts` stores an integer value.

## Task 8

In this task, you'll determine the Boolean value that represents whether the current number of login attempts a user has made is less than or equal to the maximum number of login attempts allowed.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [8]:

```
# Assign `max_logins` to the value 3

max_logins = 3

# Assign `login_attempts` to the value 2

login_attempts = 2

# Determine whether the current number of login attempts a user has made is
# less than or equal to the maximum number of login attempts allowed,
# and display the resulting Boolean value

print(login_attempts <= max_logins)
```

True

#### Hint 1

#### Hint 2

#### Question 6

##### What is the output? What does this mean?

The output above is `True`, which indicates that `login_attempts` is less than or equal to `max_logins`. In other words, the current number of attempts the user has made to log in has not yet exceeded the maximum number of attempts allowed.

## Task 9

This code continues to check for the Boolean value of whether `max_logins` is less than or equal to `login_attempts`. In this task, reassign other values to `login_attempts`. For example, you might choose a value that is higher than the maximum number of attempts allowed. Observe how the output changes.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [9]:

```
# Assign `max_logins` to the value 3

max_logins = 3

# Assign `login_attempts` to a specific value
```

```
login_attempts = 4

# Determine whether the current number of login attempts a user has made is
# less than or equal to the maximum number of login attempts allowed,
# and display the resulting Boolean value

print(login_attempts <= max_logins)
```

False

### Hint 1

### Question 7

Based on the different values you assigned to `login_attempts`, what did you observe about the output?

The Boolean value in the output changes depending on the value assigned to `login_attempts`. For example, when `login_attempts` is assigned to 4, the output is `False`, which indicates that `login_attempts` is not less than or equal to `max_logins`. In other words, the current number of log in attempts the user has made has exceeded the maximum number of attempts allowed.

## Task 10

Finally, you can also assign a Boolean value of `True` or `False` to a variable.

In this task, you'll create a variable called `login_status`, which is a Boolean that represents whether a user is logged in. Assign `False` to this variable and store its data type in a variable called `login_status_type` and display it.

Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [10]:

```
# Assign `login_status` to the Boolean value False

login_status = False

# Assign `login_status_type` to the data type of `login_status`

login_status_type = type(login_status)

# Display `login_status_type`

print(login_status_type)
```



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

Hint 1

Hint 2

### Question 8

Based on the output above, what do you observe about the data type of `login_status`?

The output above shows that the data type of the `login_status` is `bool`, which means that `login_status` stores a Boolean value.

## Conclusion

What are your key takeaways from this lab?

- There are many useful operators in Python that help you work with variables.
  - The `=` assignment operator allows you to assign or reassign a specific value to a variable.
  - The `<=` comparison operator allows you to compare the value of one variable to the value of another.
- The `type()` function in Python helps you to determine the data type of an object.
  - If you pass in a variable to `type()`, it will output the data type of the value stored in the variable.
- The `print()` function in Python allows you to display information.
  - It can take in a value directly, a variable that stores a value, or a comparison between variables that evaluates to a Boolean value.