

Kindly research and if possible try to use the Jupyter Notebook:

Strings

- Creating a String

- A string is contained within double quotes or (" "):

```
my_string = "Angelle Salvador"  
print(my_string)      #Output: Angelle Salvador
```

- You can also use single quotes or (' ')

```
my_string = 'Angelle Salvador'  
print(my_string)      #Output: Angelle Salvador
```

- A string can be spaces and digits:

For ex.

```
'1 2 3 4 5 6 '
```

- A string can also be special characters:

For ex.

```
'@#2_#]&*^%$'
```

- Accessing Characters in the String

Accessing characters in the string is the same as we do in Python. We can use indexing or numeric representation of their index position to access. Lists in Python are zero-indexed meaning that the first element in the list is at index 0, the second element is at index 1 and so on.

For ex.

```
names = ['Angelle', 'Gelle', 'Gengelle']  
print(names[0])      #Output: Angelle  
print(names[1])      #Output: Gelle  
print(names[2])      #Output: Gengelle
```

- Removing Space from a String

To remove spaces from a string, we can use these methods such as **replace()** or **split()**, or using regular expressions.

For ex.

- The string with spaces

```
my_string = "Hello, world! This is a string with spaces."
```

- Removing space from a string using the **replace() method**

```
my_string_without_spaces = my_string.replace(" ", "")
```

```
print(my_string_without_spaces)
```

```
#Output: Hello,world!Thisisastringwithspaces.
```

- Python String Methods

Here are some commonly used Python String Methods:

1. **upper()** and **lower()**: Convert a string to uppercase or lowercase.

For ex.

```
my_string = "Angelle"  
upper_case_string = my_string.upper()  
lower_case_string = my_string.lower()  
print(upper_case_string)      #Output: ANGELLE  
print(lower_case_string)     #Output: angelle
```

2. **strip():** Remove leading and trailing whitespace characters from a string.

For ex.

```
my_string = "Hello, World!"  
stripped_string = my_string.strip()  
print(stripped_string)      #Output: Hello, World!
```

3. **replace():** Replace occurrences of a substring with another substring.

For ex.

```
my_string = "Hello, Angelle!"  
replaced_string = my_string.replace("Angelle", "Gelle")  
print(replaced_string)      #Output: Hello, Gelle!
```

4. **split():** Split a string into a list of substrings based on a delimiter.

For ex.

```
my_string = "tulips, rose, peony, orchid"  
flowers_list = my_string.split(",")  
print(flowers_list)      #Output: ['tulips', 'rose', 'peony', 'orchid']
```

5. **join():** Join a list of strings into a single string using a delimiter.

For ex.

```
my_string = ['tulips', 'rose', 'peony', 'orchid']  
flowers_list = ",".join(flowers_list)  
print(flowers_list)      #Output: tulips, rose, peony, orchid]
```

6. **find() and index():** Find the index of the first occurrence of a substring within a string.

For ex.

```
my_string = "Hello, World!"  
index1 = my_string.find("World")  
index2 = my_string.index("World")  
print(index1)      #Output: 7  
print(index2)      #Output: 7
```

7. **startswith() and endswith():** Check if a string starts or ends with a given substring.

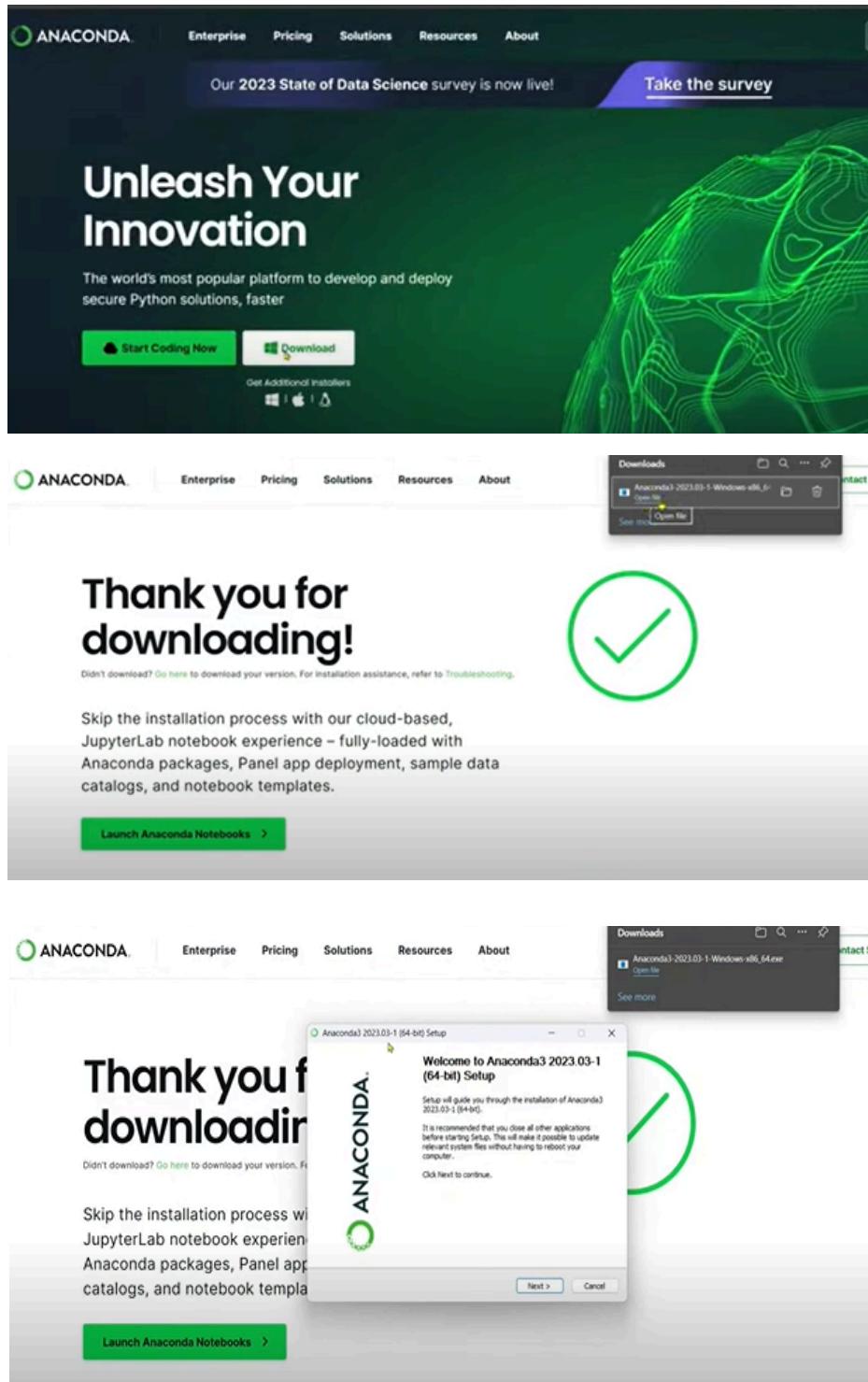
For ex.

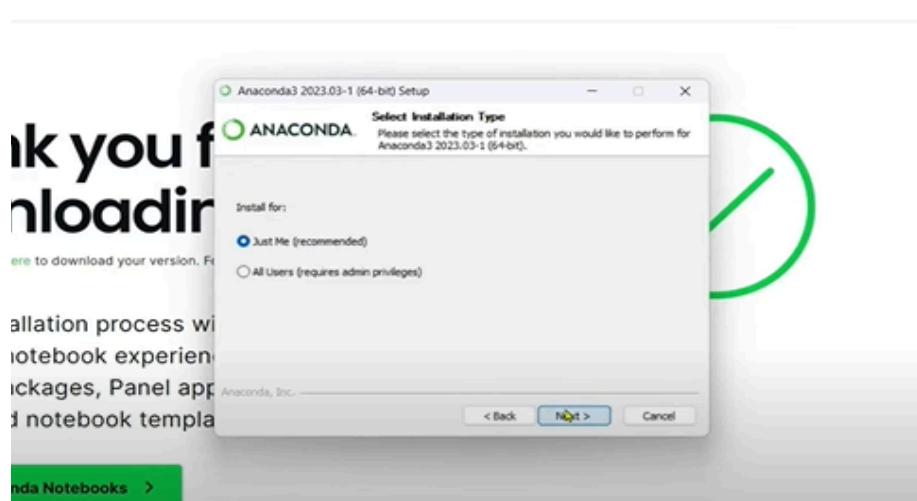
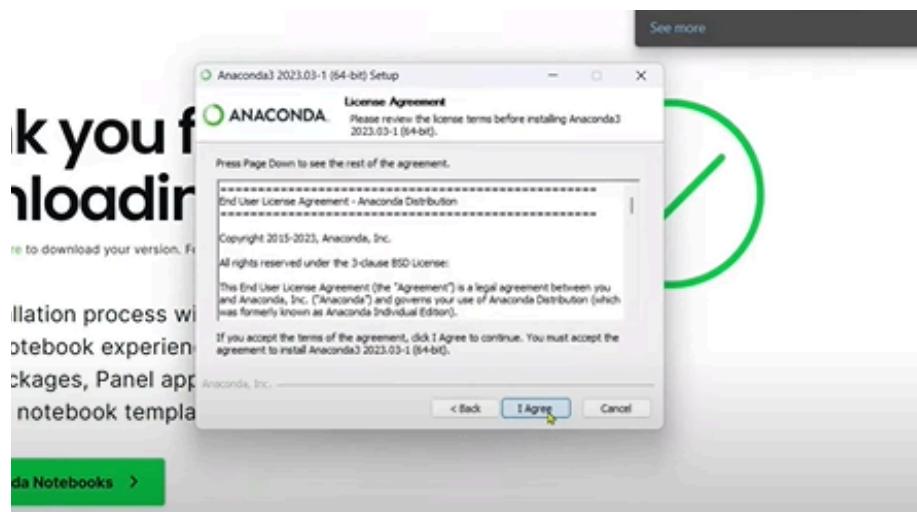
```
my_string = "Hello, World!"  
starts_with_hello = my_string.startswith("Hello")  
ends_with_world = my_string.endswith("World")  
print(starts_with_hello)      #Output: True  
print(ends_with_world)       #Output: False
```

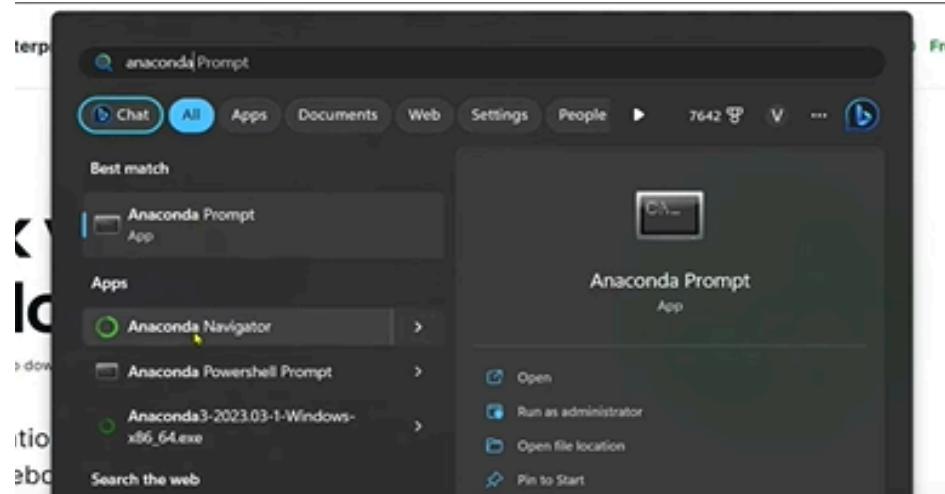
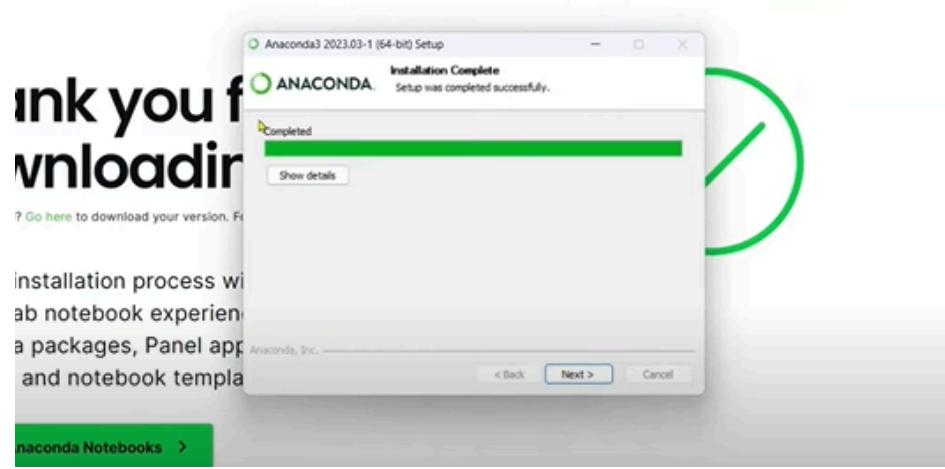
Python and jupyter notebook

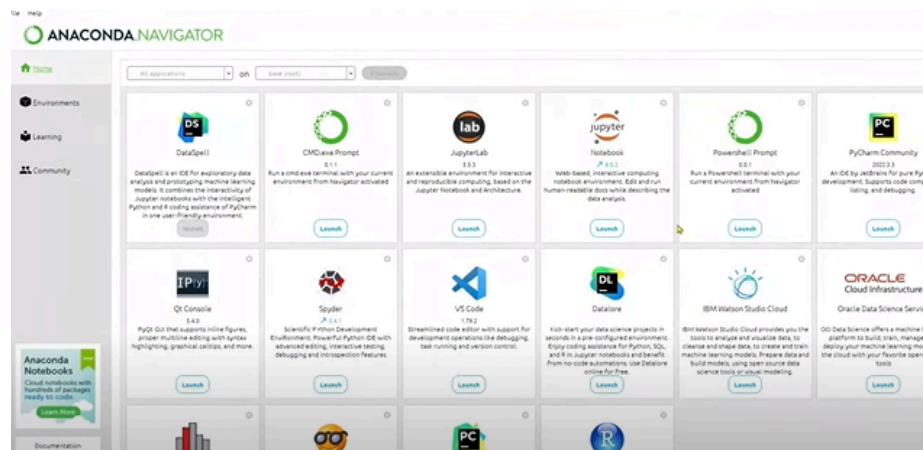
- Launch Jupyter Notebook

These are the steps:





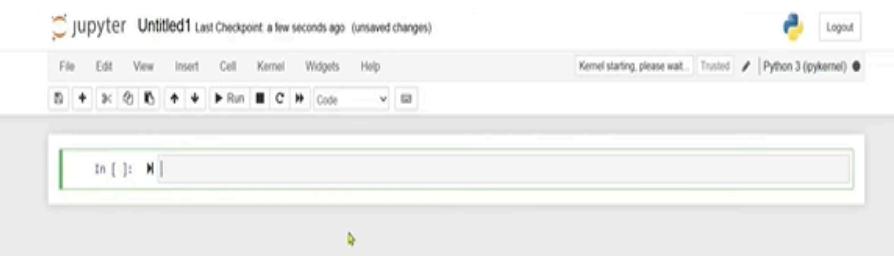




• Open a notebook file



- **Launch Jupyter Notebook**



- **Start writing a Jupyter Notebook**

