

Strings

1. Creating a String

- In Python, you can create a string by enclosing a sequence of characters within single or double quotes. For example, `str1 = "Hello"` or `str2 = 'World'`.

```
#1
# Single quotes
str1 = 'Hello'

# Double quotes
str2 = "World"

print(str1)
print(str2)
```

```
Hello
World
```

2. Accessing Characters in the String

- You can access individual characters in a string using indexing. For example, `str1[0]` would give you the string's first character stored in `str1`. Python also supports negative indexing, with `-1` referring to the last character of the string.

```
#2

str = "Python Programming"

# Indexing
print(str[0])
print(str[-1])
```

```
P
g
```

3. Removing Space from a String

- The `strip()` method can be used to remove spaces from a string. It removes any leading and trailing spaces. For example, `str3 = " Hello World"`
`".strip()` would result in `str3` being `Hello World`.

```
#3
str = "    Hello, World!    "
print(str.strip())

Hello, World!
```

4. Python String Methods

- Python provides some methods that can be used with strings. Some of them are:
 - `upper()`: Converts all characters in the string to uppercase.
 - `lower()`: Converts all characters in the string to lowercase.
 - `replace(old, new)`: Replaces all occurrences of the `old` substring with the `new` substring.
 - `split(delimiter)`: Splits the string into a list of substrings based on the provided delimiter.

```
#4
str = "Hello, world! World is big."

# Upper and Lower
print(str.upper())
print(str.lower())

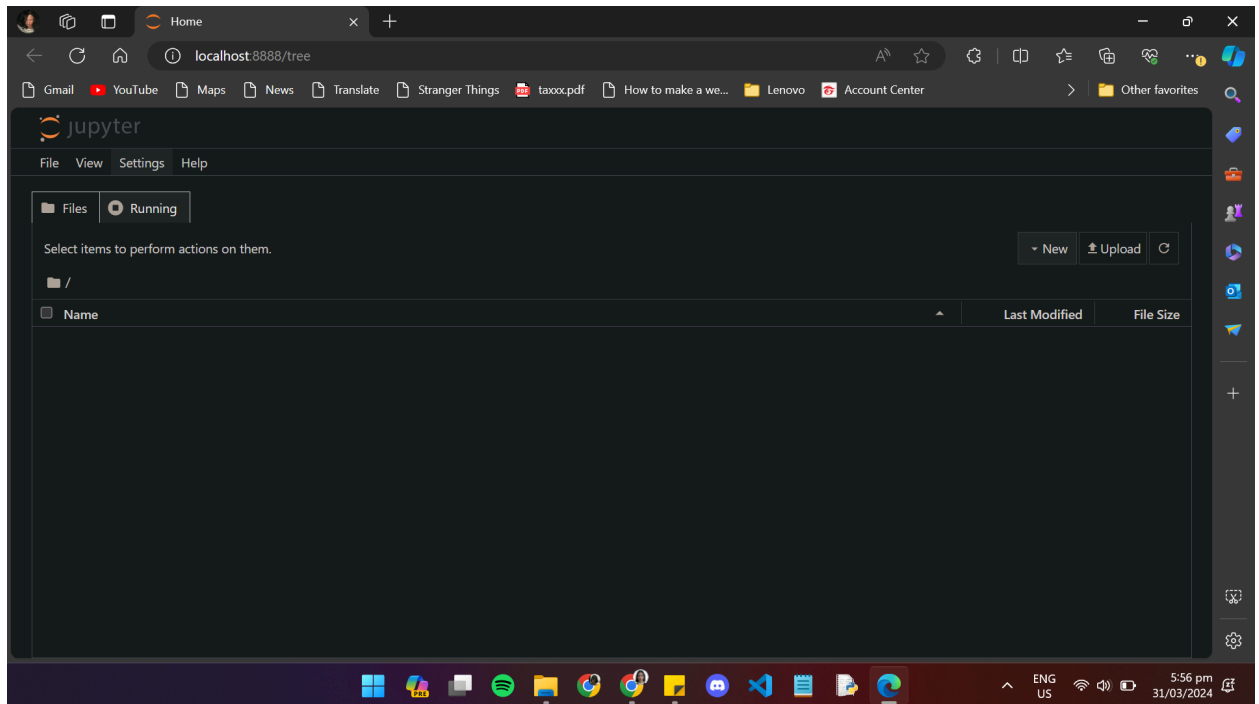
# Replace
print(str.replace("world", "Python"))

# Split
print(str.split())
```

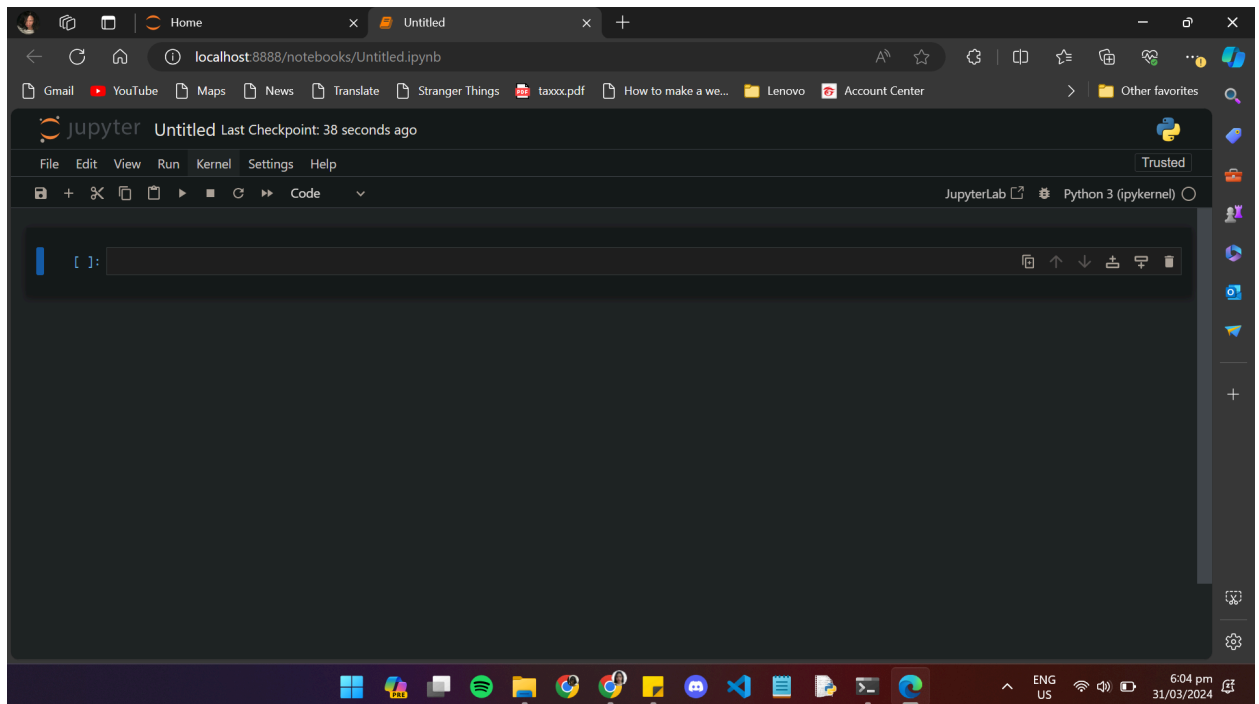
```
HELLO, WORLD! WORLD IS BIG.
hello, world! world is big.
Hello, Python! World is big.
['Hello,', 'world!', 'World', 'is', 'big.']
```

Python and jupyter notebook

- **Launch Jupyter Notebook**



- **Open a notebook file**



- **Start writing a Jupyter Notebook**

