😊 Methods in Python

A method is a function associated with an object that performs specific actions on that object. Methods are called using dot notation and can modify or return values.

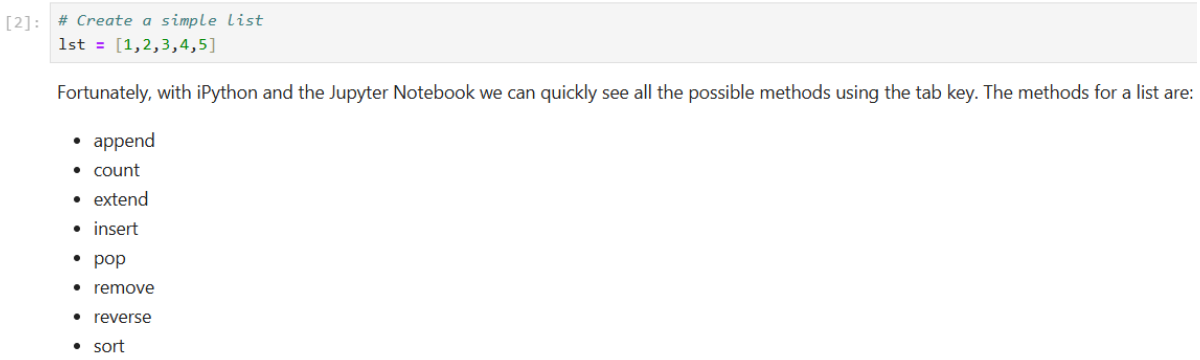
Use Cases:

- Modifying strings (e.g., converting to uppercase)
- Adding or removing elements from a list
- Retrieving keys or values from a dictionary

Benefits:

- Improves code readability
- Makes operations on objects easier
- Reduces the need for writing repetitive code

1. In the code below we just created a simple list.

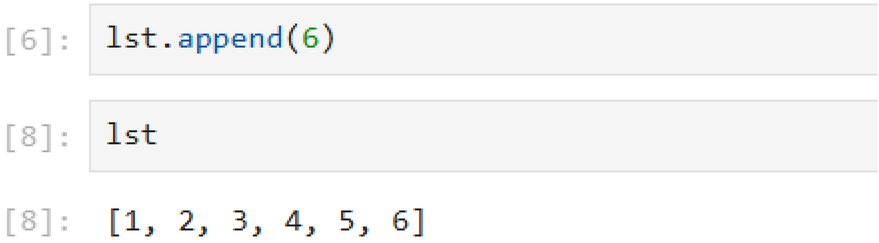


```
[2]: # Create a simple list
     lst = [1,2,3,4,5]
```

Fortunately, with iPython and the Jupyter Notebook we can quickly see all the possible methods using the tab key. The methods for a list are:

- append
- count
- extend
- insert
- pop
- remove
- reverse
- sort

2. Using the **append () method,** we have added a new element at the end of our list.

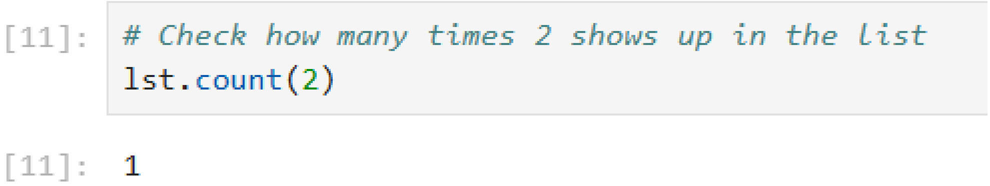


```
[6]: lst.append(6)

[8]: lst

[8]: [1, 2, 3, 4, 5, 6]
```

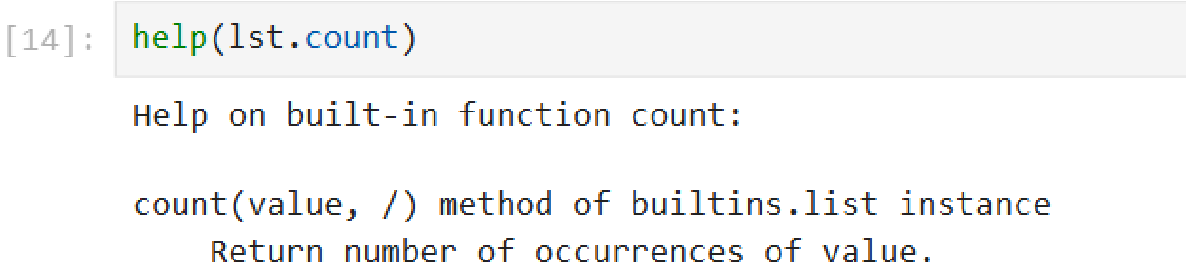
3. This time we have used the **count () method** to see how many times 2 has appeared in the list.



```
[11]: # Check how many times 2 shows up in the list
      lst.count(2)

[11]: 1
```

4. You can use the help to view the methods property and play around with other methods as well.



```
[14]: help(lst.count)

      Help on built-in function count:

      count(value, /) method of builtins.list instance
          Return number of occurrences of value.
```