

## Basic Slicing:

- \* Given the string text = "PythonProgramming", extract the substring "Python".  
text = "PythonProgramming"  
text[0:6]  
'Python'
- \* Write code to reverse the string "DataScience" using slicing.  
string="DataScience"  
string[::-1]  
'ecneicSataD'
- \* From the string sentence = "Machine Learning is fascinating", extract the substring "Learning".  
sentence = "Machine Learning is fascinating"  
sentence[8:16]  
'Learning'
- \* Given the string word = "abcdefghij", extract every second character.  
word = "abcdefghij"  
word[::2]  
'acegi'
- \* Use negative indexing to extract the last three characters from the string "DeepLearning".  
thestring= "DeepLearning"  
thestring[-3:]  
'ing'
- \* Given the string data = "0123456789", extract the substring "13579".  
data = "0123456789"  
data[1::2]  
'13579'
- \* For the string quote = "To be or not to be", extract the substring starting from "or" to the end.  
quote = "To be or not to be"  
quote[6:]  
'or not to be'
- \* Extract the first half of the string "ArtificialIntelligence" using slicing.  
string = "ArtificialIntelligence"  
len(string)  
22  
string[0:11]  
'Artificiall'
- \* Write code to extract the last half of the string "NeuralNetworks".

```
string1="NeuralNetworks"
```

```
len(string1) 14
```

```
string1[7:]
```

```
'etworks'
```

### List Slicing Questions

- \* From the list numbers = [10, 20, 30, 40, 50, 60], extract the elements from the list using inbuilt function. [30, 40].  

```
numbers = [10, 20, 30, 40, 50, 60]  
numbers[2:4]  
[30, 40]
```
- \* Write code to reverse the list fruits = ['apple', 'banana', 'cherry', 'date'] using slicing.  

```
fruits = ['apple', 'banana', 'cherry', 'date']  
fruits[::-1]  
['date', 'cherry', 'banana', 'apple']
```
- \* From the list data = [100, 200, 300, 400, 500, 600], remove the first three elements using slicing.  

```
listdata = [100, 200, 300, 400, 500, 600]  
listdata[3:]  
[400, 500, 600]
```
- \* Given the list colors = ['red', 'green', 'blue', 'yellow', 'purple'], extract the elements from 'green' to 'yellow'.  

```
colors = ['red', 'green', 'blue', 'yellow', 'purple']  
colors[1:4]  
['green', 'blue', 'yellow']
```

### Slice and Replace:

- \* Modify the list animals = ['cat', 'dog', 'rabbit', 'hamster'] by replacing 'dog' and 'rabbit' with ['lion', 'tiger'].  

```
animals = ['cat', 'dog', 'rabbit', 'hamster']  
animals[1,2]= ['lion', 'tiger']  
['cat', 'lion', 'tiger', 'hamster']
```
- Given the list scores = [75, 85, 95, 65, 55, 45], extract the last three elements using slicing.  

```
scores = [75, 85, 95, 65, 55, 45]  
scores[-3:]  
[65, 55, 45]
```

\* Write code to reverse all the elements present in the list elements of the list  
items = [10, 20, 30, 40, 50, 60, 70].

```
items = [10, 20, 30, 40, 50, 60, 70]
```

```
items[::-1]
```

```
[70, 60, 50, 40, 30, 20, 10]
```

\* From the tuple alphabets = ('a', 'b', 'c', 'd', 'e'), extract the subtuple ('b', 'c', 'd').

```
alphabets = ('a', 'b', 'c', 'd', 'e')
```

```
alphabets[1:4]
```

```
('b', 'c', 'd')
```

\* Write code to reverse the tuple digits = (0, 1, 2, 3, 4, 5) using slicing.

```
digits = (0, 1, 2, 3, 4, 5)
```

```
digits[::-1]
```

```
(5, 4, 3, 2, 1, 0)
```

\* Given the tuple months = ('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'), extract the last three months using negative indexing.

```
months = ('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun')
```

```
months[-3:]
```

```
('Apr', 'May', 'Jun')
```

\* From the tuple points = (1, 2, 3, 4, 5, 6), extract every second element.

```
points = (1, 2, 3, 4, 5, 6)
```

```
points[::2]
```

```
(1, 3, 5)
```

\* Given the tuple sequence = (0, 1, 2, 3, 4, 5, 6, 7, 8), extract the subtuple (0, 3, 6) using slicing.

```
sequence = (0, 1, 2, 3, 4, 5, 6, 7, 8)
```

```
sequence[::3]
```

```
(0, 3, 6)
```

\* Extract the first three elements from the tuple weekdays = ('Mon', 'Tue', 'Wed', 'Thu', 'Fri') and reverse them.

\* weekdays = ('Mon', 'Tue', 'Wed', 'Thu', 'Fri')

```
1. a=weekdays[:3]
```

```
a[::-1]
```

```
2. weekdays[2::-1]
```

```
('Wed', 'Tue', 'Mon')
```

\* From the tuple languages = ('Python', 'Java', 'C++', 'Ruby', 'JavaScript'), remove the first and last elements.

```
languages = ('Python', 'Java', 'C++', 'Ruby', 'JavaScript')
```

```
languages[1:-1]
```

```
('Java', 'C++', 'Ruby')
```

## \* LIST

\* Given the list numbers = [10, 20, 30, 40, 50, 60, 70], write a Python script to:

\* Extract the first three elements.

```
numbers1 = [10, 20, 30, 40, 50, 60, 70]
```

```
Python=[10, 20, 30, 40, 50, 60, 70]
```

```
Python[:3]
```

```
[10, 20, 30]
```

\* Extract the last three elements.

```
Python[-3:]
```

```
[50, 60, 70]
```

\* Extract every second element starting from index 1.

```
Python[1::2]
```

```
[20, 40, 60]
```

\* Reverse the list .

```
Python[::-1]
```

```
[70, 60, 50, 40, 30, 20, 10]
```

\* Given the list fruits = ['apple', 'banana', 'cherry', 'date', 'fig', 'grape'], replace the middle two elements with ['mango', 'kiwi'] using slicing.

```
fruits = ['apple', 'banana', 'cherry', 'date', 'fig', 'grape']
```

```
fruits[2:4]=['mango','kiwi']
```

```
fruits
```

```
['apple', 'banana', 'mango', 'kiwi', 'fig', 'grape']
```

\* Given a nested list matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]], write code to extract the second column (i.e., [2, 5, 8]) using list comprehension and slicing.

```
m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
a=[]
```

```
a.extend([m[0][1],m[1][1],m[2][1]])
```

```
[2, 5, 8]
```

\* Given the list colors = ['red', 'green', 'blue'], write a Python script to: \* Add 'yellow' at the end of the list.

```
colors = ['red', 'green', 'blue']
```

```
colors.append('yellow')
```

```
colors
```

```
['red', 'green', 'blue', 'yellow']
```

\*insert 'orange' at the second position.

```
colors.insert(1,'orange')
```

```
colors
```

```
['red', 'orange', 'green', 'blue', 'yellow']
```

\* Extend the list with another list ['purple', 'pink'].

```
colors.extend(['purple', 'pink'] )
```

```
colors
```

```
['red', 'orange', 'green', 'blue', 'yellow', 'purple', 'pink']
```

\* Given the list animals = ['cat', 'dog', 'rabbit', 'hamster'], perform the following tasks: \*  
Remove the last element from the list.

```
animals = ['cat', 'dog', 'rabbit', 'hamster']
```

```
animals.remove('hamster')
```

```
animals
```

```
['cat', 'dog', 'rabbit']
```

\* Remove 'dog' using an inbuilt function.

```
animals.remove('dog')
```

```
animals
```

```
['cat', 'rabbit']
```

\* Delete the first element using slicing.

```
animals[1:]
```

```
['rabbit']
```

## SET

Adding Values to a Set:

Given the set even\_numbers = {2, 4, 6, 8}, write code to:

```
even_numbers = {2, 4, 6, 8}
```

```
even_numbers
```

```
{8, 2, 4, 6}
```

Add the number 10 to the set.

```
even_numbers.add(10)
```

```
even_numbers
```

```
{2, 4, 6, 8, 10}
```

Add multiple numbers [12, 14] to the set.

```
even_numbers.update([12, 14])
```

```
even_numbers
```

```
{2, 4, 6, 8, 10, 12, 14}
```

Removing Values from a Set:

Given the set prime\_numbers = {2, 3, 5, 7, 11}, write a Python script to:

```
numbers = {2, 3, 5, 7, 11}
```

```
numbers
```

```
{2, 3, 5, 7, 11}
```

Remove the number 7 using an inbuilt function.

```
numbers.remove(7)
```

```
numbers
```

```
{2, 3, 5, 11}
```

Attempt to remove 13 and handle any exceptions if the element is not found.

```
numbers.remove(13)
```

```
Traceback (most recent call last):
```

```
File "<pyshell#150>", line 1, in <module>
```

```
    numbers.remove(13)
```

```
KeyError: 13
```

## Dictionary

Adding Key-Value Pairs to a Dictionary:

Given the dictionary `student_grades = {'John': 'A', 'Emily': 'B'}`, write code to:

```
student_grades = {'John': 'A', 'Emily': 'B'}
```

```
student_grades
```

```
{'John': 'A', 'Emily': 'B'}
```

Add a new student 'Sophia' with grade 'A+'.

```
student_grades['Sophia']='A+'
```

```
student_grades
```

```
{'John': 'A', 'Emily': 'B', 'Sophia': 'A+'}
```

Update 'Emily''s grade to 'A'.

```
student_grades['Emily']='A'
```

```
student_grades
```

```
{'John': 'A', 'Emily': 'A', 'Sophia': 'A+'}
```

Deleting Key-Value Pairs from a Dictionary:

```
student_grades.popitem()
```

```
('Sophia', 'A+')
```

Given the dictionary inventory = {'apple': 5, 'banana': 3, 'orange': 2}, perform the following tasks:

```
inventory = {'apple': 5, 'banana': 3, 'orange': 2}
```

```
inventory
```

```
{'apple': 5, 'banana': 3, 'orange': 2}
```

Remove the key 'banana'.

```
inventory.pop('banana')
```

```
3
```



inventory

{'apple': 5, 'orange': 2}

Clear all items from the dictionary.

inventory.clear()

inventory

{}