

✓ Strings in Python (Parth Manoj Gajare AIDS - 15)

```
# Assigning string to a variable
a = 'This is a string'
print (a)
b = "This is a string"
print (b)
c= '''This is a string'''
print (c)
```

```
↩ This is a string
  This is a string
  This is a string
```

✓ Lists in Python

```
# Declaring a list
L = [1, "a" , "string" , 1+2]
print (L)
#Adding an element in the list
L.append(6)
print (L)
#Deleting last element from a list
L.pop()
print (L)
#Displaying Second element of the list
print (L[1])
```

```
↩ [1, 'a', 'string', 3]
  [1, 'a', 'string', 3, 6]
  [1, 'a', 'string', 3]
  a
```

✓ Tuples in Python

```
tup = (1, "a", "string", 1+2)
print(tup)
print(tup[1])
```

```
↩ (1, 'a', 'string', 3)
  a
```

✓ Dictionaries in Python

A Python dictionary is a data structure that stores the value in key: value pairs. Values in a dictionary can be of any data type and can be duplicated, whereas keys can't be repeated and must be immutable.

```
d = {1: 'Lorem', 2: 'Ipsum', 3: 'Dolerum'}
print(d)
```

```
↩ {1: 'Lorem', 2: 'Ipsum', 3: 'Dolerum'}
```

✓ Create a Dictionary

```
# create dictionary using { }
d1 = {1: 'Game', 2: 'of', 3: 'Thrones'}
print(d1)
```

```
# create dictionary using dict() constructor
d2 = dict(a = "House", b = "of", c = "Cards")
print(d2)
```

```
↩ {1: 'Game', 2: 'of', 3: 'Thrones'}
  {'a': 'House', 'b': 'of', 'c': 'Cards'}
```

✓ Accessing Dictionary Items

```
d = { "name": "Alice", 1: "Python", (1, 2): [1,2,4] }
```

```
# Access using key
```

```
print(d["name"])
```

```
# Access using get()
print(d.get("name"))
```

```
↗ Alice
Alice
```

✓ Adding and Updating Dictionary Items

```
d = {1: 'Game', 2: 'of', 3: 'Thrones'}
```

```
# Adding a new key-value pair
d["age"] = 22
```

```
# Updating an existing value
d[1] = "Python dict"
```

```
print(d)
```

```
↗ {1: 'Python dict', 2: 'of', 3: 'Thrones', 'age': 22}
```

✓ Delete a Specific Item Using del

```
# Sample dictionary
my_dict = {
    "name": "John",
    "age": 25,
    "city": "New York"
}
```

```
# Print original dictionary
print("Original Dictionary:", my_dict)
```

```
# Delete an item by its key
del my_dict["age"]
```

```
# Print updated dictionary
print("Updated Dictionary:", my_dict)
```

```
↗ Original Dictionary: {'name': 'John', 'age': 25, 'city': 'New York'}
Updated Dictionary: {'name': 'John', 'city': 'New York'}
```

✓ Delete Using .pop() Method

```
# Sample dictionary
my_dict = {
    "name": "John",
    "age": 25,
    "city": "New York"
}
```

```
# Print original dictionary
print("Original Dictionary:", my_dict)
```

```
# Remove an item and capture the removed value
removed_value = my_dict.pop("city")
```

```
# Print updated dictionary and the removed value
print("Updated Dictionary:", my_dict)
print("Removed Value:", removed_value)
```

```
↗ Original Dictionary: {'name': 'John', 'age': 25, 'city': 'New York'}
Updated Dictionary: {'name': 'John', 'age': 25}
Removed Value: New York
```

✓ Delete All Items Using .clear()

```
# Sample dictionary
my_dict = {
    "name": "John",
    "age": 25,
    "city": "New York"
}
```

```
}

# Print original dictionary
print("Original Dictionary:", my_dict)

# Clear all items from the dictionary
my_dict.clear()

# Print the empty dictionary
print("Cleared Dictionary:", my_dict)
```

↔ Original Dictionary: {'name': 'John', 'age': 25, 'city': 'New York'}
Cleared Dictionary: {}

Start coding or [generate](#) with AI.