TOPS TECHNOLOGY

Python – Collections, functions and Modules

Presented By:
Nandni Vala

Dictionaries

1.Introduction to dictionaries: key-value pairs.

- A **dictionary** in Python is a collection of **key-value pairs**:
- **Key**: A unique identifier used to access a value.
- **Value**: The data associated with a key.
- **Key Features**
- **Unordered**: The order of items is not guaranteed.
- **Mutable**: You can change, add, or remove key-value pairs.
- **Unique Keys**: Keys must be unique within a dictionary.
- **Syntax**:
- my_dict = {"key1": "value1", "key2": "value2"}

- **Example** :

```
person = {
    "name": "John",
    "age": 30,
    "city": "New York"
}
print(person["name"])
Output: John
```

2.Accessing, adding, updating, and deleting dictionary elements

- **Accessing Dictionary Elements :**
- Access elements using their **key**.
- **Example** :

-
  ```
  person = {
   "name": "John",
   "age": 30,
   "city": "New York"
  }
  ```

-
  ```
  print(person["name"])
  ```
- Output: John

- **Adding New Elements :**
- Add a new key-value pair by simply assigning a value to a new key.
- person["email"] = john@example.com
- print(person)
- Output: {'name': 'John', 'age': 30, 'city': 'New York', 'email': 'john@example.com'}
- **Updating Elements :**
- Update the value of an existing key by assigning a new value.
- person["age"] = 31 # Update the age value
- print(person)
- Output: {'name': 'John', 'age': 31, 'city': 'New York', 'email': 'john@example.com'}

- **Deleting Elements :**
- **Using del**: Removes a key-value pair by key.
- del person["city"]
- print(person)
- Output: {'name': 'John', 'age': 31, 'email': 'john@example.com'}
- **Using pop()**: Removes the key-value pair and returns the value.
- removed_email = person.pop("email")
- print(removed_email)
- print(person)
- Output: {'name': 'John', 'age': 31}
- **Using clear()**: Removes all key-value pairs from the dictionary.
- person.clear()
- print(person)

3.Dictionary methods like keys(), values(), and items().

- **keys() Method :**
- Returns a **view object** that displays all the keys in the dictionary.
- person = {
-     "name": "John",
-     "age": 30,
-     "city": "New York"
- }
- keys = person.keys()
- print(keys)
- Output: dict_keys(['name', 'age', 'city'])

- **values() Method :**
- Returns a **view object** that displays all the values in the dictionary.
- values = person.values()
- print(values)
- Output: dict_values(['John', 30, 'New York'])
- **items() Method :**
- Returns a **view object** that displays all key-value pairs as tuples.
- items = person.items()
- print(items)
- Output: dict_items([('name', 'John'), ('age', 30), ('city', 'New York')])