Lecture 24

Dictionary part 2

In [ ]:
```
Imagine you are managing an inventory of products in a store.
Create a Python program that initializes a dictionary
to store the following information about products:

Product ID as the key (an integer).
Product name as the value (a string).
Your program should:

Create an empty dictionary.
Prompt the user to enter details for at least 3 products.
Populate the dictionary with the entered product IDs and names.
Display the final dictionary containing all product IDs and names.
```

In [1]:
```
def create_product_inventory(num_products):
    inventory = {}

    for i in range(1, num_products + 1):
        product_id = int(input(f"Enter product ID {i}: "))
        product_name = input(f"Enter product name for ID {product_id}: ")
        inventory[product_id] = product_name

    return inventory

# Prompt the user for the number of products
num_products = int(input("Enter the number of products: "))

# Create the product inventory dictionary
inventory_dict = create_product_inventory(num_products)

# Display the product inventory
print("\nProduct Inventory:")
print(inventory_dict)
```

```
Enter the number of products: 3
Enter product ID 1: 101
Enter product name for ID 101: shampoo
Enter product ID 2: 102
Enter product name for ID 102: hair product
Enter product ID 3: 103
Enter product name for ID 103: cosmetics

Product Inventory:
{101: 'shampoo', 102: 'hair product', 103: 'cosmetics'}
```

Characterstics of the dictionary

ASESSING VALUES USING KEYS

In [2]:
```
person = {
    'name':"swati",
    'age':30,
    'city':"newyork"
}
```

In [3]:
```
print(person['name'])
```

```
swati
```

In [4]:
```
print(person['age'])
```

```
30
```

In [5]:
```
print(person['city'])
```

```
newyork
```

```
Dictionaries in Python do not maintain any specific order for their
elements. The order in which items are stored may not necessarily be the
order in which they are retrieved. From Python 3.7 onwards, dictionaries
maintain insertion order, which means items are generally stored and
returned in the order they were added.
```

In [9]:
```
person = {
    'name':'swati',
    'age':30
}
```

In [10]:
```
print(person)
```

```
{'name': 'swati', 'age': 30}
```

In [11]:
```
person = {
    'name':'swati',
    'age':30,
    'name':'nishant'
}
```

In [12]:
```
print(person)
```

```
{'name': 'nishant', 'age': 30}
```

Flexible key types

In [ ]:
```
While keys in dictionaries must be unique, they can be
of various data types, including strings, integers,
floats, and tuples (if they contain only hashable objects).
```

In [15]:
```
data = {
    'name':'john',
    123:"integer key",
    (1,2):"tuple key",
    3.6:'float key'
}
```

In [16]:
```
print(data)
```

```
{'name': 'john', 123: 'integer key', (1, 2): 'tuple key', 3.6: 'float key'}
```

In [17]:
```
data = {
    'name':'john',
    123:"integer key",
    (1,2):"tuple key",
    3.6:'float key',
    [1,2,3]:"list key"
}
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[17], line 1
----> 1 data = {
      2     'name':'john',
      3     123:"integer key",
      4     (1,2):"tuple key",
      5     3.6:'float key',
      6     [1,2,3]:"list key"
      7 }

TypeError: unhashable type: 'list'
```

In [18]:
```
data = {
    'name':'john',
    123:"integer key",
    (1,2):"tuple key",
    3.6:'float key',
    {'age':30, 'year':1997}:"dictionary key"
}
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 data = {
      2     'name':'john',
      3     123:"integer key",
      4     (1,2):"tuple key",
      5     3.6:'float key',
      6     {'age':30, 'year':1997}:"dictionary key"
      7 }

TypeError: unhashable type: 'dict'
```

Duplicates not allowed

In [19]:
```
dict = {
    'ID':'123ABQ',
    'color':'red',
    'year':1964,
    'year':2000
}
```

In [20]:
```
print(dict)
```

```
{'ID': '123ABQ', 'color': 'red', 'year': 2000}
```

In [ ]:
```

```