Creating a Dictionary

In [ ]:

```
# Creating an empty Dictionary
Dict = {}
print("Empty Dictionary: ")
print(Dict)
```

In [2]:

```
# Creating a Dictionary
# with Integer Keys
Dict = {1: 'Geeks', 2: 'For', 3: 'Geeks'}
print("\nDictionary with the use of Integer Keys: ")
print(Dict)
```

```
Dictionary with the use of Integer Keys:
{1: 'Geeks', 2: 'For', 3: 'Geeks'}
```

In [3]:

```
# Creating a Dictionary
# with Mixed keys
Dict = {'Name': 'Geeks', 1: [1, 2, 3, 4]}
print("\nDictionary with the use of Mixed Keys: ")
print(Dict)
```

```
Dictionary with the use of Mixed Keys:
{'Name': 'Geeks', 1: [1, 2, 3, 4]}
```

In [4]:

```
# Creating a Dictionary
# with dict() method
Dict = dict({1: 'Geeks', 2: 'For', 3:'Geeks'})
print("\nDictionary with the use of dict(): ")
print(Dict)
```

```
Dictionary with the use of dict():
{1: 'Geeks', 2: 'For', 3: 'Geeks'}
```

In [5]:

```
# Creating a Dictionary
# with each item as a Pair
Dict = dict([(1, 'Geeks'), (2, 'For')])
print("\nDictionary with each item as a pair: ")
print(Dict)
```

```
Dictionary with each item as a pair:
{1: 'Geeks', 2: 'For'}
```

In [6]:

```
# Creating a Nested Dictionary
# as shown in the below image
Dict = {1: 'Geeks', 2: 'For',
            3:{'A' : 'Welcome', 'B' : 'To', 'C' : 'Geeks'}}

print(Dict)
```

```
{1: 'Geeks', 2: 'For', 3: {'A': 'Welcome', 'B': 'To', 'C': 'Geeks'}}
```

Adding elements to a Dictionary

In Python Dictionary, Addition of elements can be done in multiple ways. One value at a time can be added to a Dictionary by defining value along with the key e.g. Dict[Key] = ‘Value’. Updating an existing value in a Dictionary can be done by using the built-in update() method. Nested key values can also be added to an existing Dictionary. Note- While adding a value, if the key value already exists, the value gets updated otherwise a new Key with the value is added to the Dictionary.

In [7]:

```
# Creating an empty Dictionary
Dict = {}
print("Empty Dictionary: ")
print(Dict)
```

```
Empty Dictionary:
{}
```

In [8]:

```
# Adding elements one at a time
Dict[0] = 'Geeks'
Dict[2] = 'For'
Dict[3] = 1
print("\nDictionary after adding 3 elements: ")
print(Dict)
```

```
Dictionary after adding 3 elements:
{0: 'Geeks', 2: 'For', 3: 1}
```

In [9]:

```
# Adding set of values
# to a single Key
Dict['Value_set'] = 2, 3, 4
print("\nDictionary after adding 3 elements: ")
print(Dict)
```

```
Dictionary after adding 3 elements:
{0: 'Geeks', 2: 'For', 3: 1, 'Value_set': (2, 3, 4)}
```

In [11]:

```
type(Dict['Value_set'])
```

Out[11]:

```
tuple
```

In [12]:

```
# Updating existing Key's Value
Dict[2] = 'Welcome'
print("\nUpdated key value: ")
print(Dict)
```

```
Updated key value:
{0: 'Geeks', 2: 'Welcome', 3: 1, 'Value_set': (2, 3, 4)}
```

In [13]:

```
# Adding Nested Key value to Dictionary
Dict[5] = {'Nested' :{'1' : 'Life', '2' : 'Geeks'}}
print("\nAdding a Nested Key: ")
print(Dict)
```

```
Adding a Nested Key:
{0: 'Geeks', 2: 'Welcome', 3: 1, 'Value_set': (2, 3, 4), 5: {'Nested':
{'1': 'Life', '2': 'Geeks'}}}
```

In [16]:

```
Dict[5]['Nested']['1']
```

Out[16]:

```
'Life'
```

Accessing elements from a Dictionary

In order to access the items of a dictionary refer to its key name.Key can be used inside square brackets.There is also a method called get() that will also help in acessing the element from a dictionary.

In [17]:

```
# Python program to demonstrate
# accesing a element from a Dictionary

# Creating a Dictionary
Dict = {1: 'Geeks', 'name': 'For', 3: 'Geeks'}

# accessing a element using key
print("Accessing a element using key:")
print(Dict['name'])
```

```
Accessing a element using key:
For
```

In [18]:

```
# accessing a element using key
print("Acessing a element using key:")
print(Dict[1])
```

```
Acessing a element using key:
Geeks
```

In [19]:

```
# accessing a element using get()
# method
print("Acessing a element using get:")
print(Dict.get(3))
```

```
Acessing a element using get:
Geeks
```

In [20]:

```
Dict['1'] = 'Learning Python'
```

In [21]:

```
Dict
```

Out[21]:

```
{1: 'Geeks', 'name': 'For', 3: 'Geeks', '1': 'Learning Python'}
```

Removing Elements from Dictionary

In Python Dictionary, deletion of keys can be done by using the del keyword. Using del keyword, specific values from a dictionary as well as whole dictionary can be deleted. Other functions like pop() and popitem() can also be used for deleting specific values and arbitrary values from a Dictionary. All the items from a dictionary can be deleted at once by using clear() method. Items in a Nested dictionary can also be deleted by using del keyword and providing specific nested key and particular key to be deleted from that nested Dictionary. Note- del Dict will delete the entire dictionary and hence printing it after deletion will raise an Error.

In [23]:

```
# Initial Dictionary
Dict = { 5 : 'Welcome', 6 : 'To', 7 : 'Geeks',
                'A' : {1 : 'Geeks', 2 : 'For', 3 : 'Geeks'},
                'B' : {1 : 'Geeks', 2 : 'Life'}}
print("Initial Dictionary: ")
print(Dict)
```

```
Initial Dictionary:
{5: 'Welcome', 6: 'To', 7: 'Geeks', 'A': {1: 'Geeks', 2: 'For', 3: 'Geek
s'}, 'B': {1: 'Geeks', 2: 'Life'}}
```

In [24]:

```
# Deleting a Key value
del Dict[6]
print("\nDeleting a specific key: ")
print(Dict)
```

```
Deleting a specific key:
{5: 'Welcome', 7: 'Geeks', 'A': {1: 'Geeks', 2: 'For', 3: 'Geeks'}, 'B':
{1: 'Geeks', 2: 'Life'}}
```

In [25]:

```
# Deleting a Key from
# Nested Dictionary
del Dict['A'][2]
print("\nDeleting a key from Nested Dictionary: ")
print(Dict)
```

```
Deleting a key from Nested Dictionary:
{5: 'Welcome', 7: 'Geeks', 'A': {1: 'Geeks', 3: 'Geeks'}, 'B': {1: 'Geek
s', 2: 'Life'}}
```

In [26]:

```
# Deleting a Key
# using pop()
Dict.pop(5)
print("\nPopping specific element: ")
print(Dict)
```

```
Popping specific element:
{7: 'Geeks', 'A': {1: 'Geeks', 3: 'Geeks'}, 'B': {1: 'Geeks', 2: 'Life'}}
```

In [28]:

```
# Deleting an arbitrary Key-value pair
# using popitem()
Dict.popitem()
print("\nPops an arbitrary key-value pair: ")
print(Dict)
```

```
Pops an arbitrary key-value pair:
{7: 'Geeks'}
```

In [29]:

```
# Deleting entire Dictionary
Dict.clear()
print("\nDeleting Entire Dictionary: ")
print(Dict)
```

```
Deleting Entire Dictionary:
{}
```

In [30]:

```
['data', 'science', 1, 2, 3, 7.5]
```

Out[30]:

```
['data', 'science', 1, 2, 3, 7.5]
```

In [31]:

```
dict_new = {'Key1':1, 'Key2':2}
```

In [33]:

```
dict_new['Key2']
```

Out[33]:

```
2
```