

গত পর্বে আমরা আলোচনা করেছিলাম মিউটেবল টাইপের ডাটা সিকোয়েন্স লিস্ট নিয়ে। এ পর্বে আলোচনা করব ডিকশনারি (dict) আর ইমিউটেবল ডাটা সিকোয়েন্স টিউপল (tuple) নিয়ে। প্রথমে আমরা টিউপল দেখব।

টিউপল সাধারণত অপরিবর্তনীয় ডাটা ধারণ করতে ব্যবহার করা হয়। টিউপল ইনিশিয়লাইজ করার কয়েকটি পদ্ধতি আছে। এম্পটি টিউপল অর্থাৎ যার কোনো উপাদান নেই, এমন টিউপল বানাতে চাইলে ()



ব্যবহার করতে হবে। শুধু একটি উপাদান নিয়ে করতে হলে শেষে ',' (কমা) যোগ করতে হবে। একাধিক উপাদান নিয়ে টিউপল তৈরি করতে চাইলে প্রত্যেকটি উপাদান 'কমা' দিয়ে আলাদা করে নিতে হবে। এছাড়া tuple() বা tuple(iterator) বিল্টইন ফাংশন ব্যবহার করে টিউপল তৈরি করা যায়।

```
>>> tuple1 = ()
>>> tuple2 = (1,)
>>> tuple3 = (1,2,3)
>>> tuple4 = tuple('abc')
>>> tuple4
('a', 'b', 'c')
```

লিস্টের ক্ষেত্রে আমরা যেমন দেখেছি যে নতুন নতুন উপাদান যোগ করা যায়, টিউপলের ক্ষেত্রে সরাসরি তা করা সম্ভব নয়। টিউপলের সাথে শুধু টিউপল যোগ করা যায়।

```
>>> tuple2._add_(tuple3)
(1, 1, 2, 3)
```

টিউপলে কোন উপাদান কতবার আছে তা দেখতে count() ফাংশন ব্যবহার করা হয়। আর কোন উপাদান প্রথম কোন ইনডেক্সে আছে এটা জানতে index() ফাংশন ব্যবহার হয়।

```
>>> t = (1,1,2,3,4,3,5,4,6,7,7,3,1)
>>> t.count(1)
3
```

```
>>> t.index(7)
9
```

এছাড়া টিউপলের আরও যেসব ফাংশন আছে, তা দেখতে IDLE-তে help(tuple) লিখে এন্টার দিলে টিউপলের সব প্রপার্টি দেখাবে। এখান থেকে প্রয়োজনীয় ফাংশন নিয়ে কাজ করা যায়।

এখন আমরা ডিকশনারি (dict) সম্পর্কে জানব। ডিকশনারি নাম থেকেই এর কাজের ধরনটা আন্দাজ করা যায়। ডিকশনারিতে শব্দ সাজানো থাকে কি-ভ্যালু পেয়ার হিসেবে, অর্থাৎ একদিকে থাকে একটি কি বা মূল শব্দ, অপরদিকে



থাকে এর ব্যাখ্যা। ডিকশনারি টাইপের ডাটাও এভাবেই কাজ করে। ডিকশনারির কনস্ট্রাক্টর মূলত তিনটি-

```
class dict(**kwargs)
class dict(mapping,
**kwargs)
class dict(iterable,
**kwargs)
চলুন দেখে নেয়া যাক
ডিকশনারির কিছু উদাহরণ-
>>> a = dict(one=1,two=2,three=3)
>>> b = {'one': 1,'two': 2,'three': 3}
>>> c = dict(zip(['one','two'],'three'),[1,2,3])
>>> d = dict([('one',1),('two',2),('three',3)])
>>> e = dict({'one': 1,'two': 2,'three': 3})
>>> a==b==c==d==e
True
```

এখানে বিভিন্নভাবে ডিকশনারি বানানো হয়েছে। a==b==c==d==e-এর ফলাফল দেখাচ্ছে True, অর্থাৎ ভিন্ন ভিন্নভাবে ডিকশনারি বানালেও সব একই।

এবার ডিকশনারির কিছু বিল্টইন ফাংশন দেখা যাক-

```
>>> d = {'three': 3, 'one': 1, 'two': 2}
>>> d['four']=4
```

এখানে বলা হয়েছে d ডিকশনারির 'four' কি-তে মান বসাতে হবে 8।

```
>>> d
{'three': 3, 'one': 1, 'four': 4, 'two': 2}
>>> del d['one']
d ডিকশনারি থেকে 'one' কি-এর মান ডিলিট করতে হবে।
```

```
>>> d
{'three': 3, 'four': 4, 'two': 2}
>>> 'four' in d
True
```

যদি 'four' ডিকশনারিতে থেকে থাকে তাহলে সত্য হবে।

```
>>> 'one' not in d
True
```

এটি আগের ফাংশনের উল্টো, অর্থাৎ যদি কি ডিকশনারিতে না থাকে তাহলে সত্য দেখাবে।

```
>>> d.clear()
```

এই ফাংশনের মাধ্যমে ডিকশনারির সব উপাদান ডিলিট করে দেয়া হয়।

```
>>> d
{}
অর্থাৎ এখন ডিকশনারিতে কোনো উপাদান নেই।
```

```
>>> d = a.copy()
d তে a-এর একটি প্রতিলিপি রাখা হচ্ছে।
```

```
>>> d
{'three': 3, 'two': 2, 'one': 1}
>>> d.get('one')
1
```

get() ফাংশন ডিকশনারির কি-এর ভ্যালু রিটার্ন করে।

```
>>> d.get('five')
>>> d.get('five',-1)
-1
```

যদি কি ডিকশনারিতে না থাকে তাহলে ডিফল্ট ভ্যালু রিটার্ন করে, সাধারণত ডিফল্ট হিসেবে none সেট করা থাকে, কিন্তু ইউজার চাইলে অন্য ভ্যালু সেট করে দিতে পারেন উপরের উদাহরণের মতো।

```
>>> d.pop('two')
2
```

pop() ফাংশন কি-এর ভ্যালুটি রিটার্ন করার সাথে সাথে ডিকশনারি থেকে ডিলিট করে দেয়।

```
>>> d
{'three': 3, 'one': 1}
>>> d = a.copy()
>>> d
{'three': 3, 'two': 2, 'one': 1}
>>> d.popitem()
('three', 3)
>>> d.popitem()
```

('one', 1) ফাংশনটি popitem() ফাংশনটি ডিকশনারি থেকে র্যান্ডমভাবে কি-ভ্যালু পেয়ার ডিলিট করে।

```
>>> d.setdefault('one')
1
>>> d.setdefault('four')
>>> d.setdefault('four',4)
4
>>> d
{'three': 3, 'one': 1, 'four': 4, 'two': 2}
```

setdefault() ফাংশনে কি দেয়া হলে যদি সেটি ডিকশনারিতে থাকে, তাহলে তার ভ্যালু রিটার্ন করে। ডিকশনারিতে না থাকলে এই কি-তে ডিফল্ট ভ্যালু সেট করে নেয়। ডিফল্ট হিসেবে none থাকলেও ইউজার নিজের কোনো ভ্যালু ডিফল্ট হিসেবে ব্যবহার করতে পারেন উপরের উদাহরণের মোট করে।

```
>>> keys = d.keys()
>>> keys
dict_keys(['two', 'one', 'three'])
```

keys() ফাংশনের মাধ্যমে ডিকশনারির কিগুলো কি-অবজেক্ট হিসেবে পাওয়া যায়।

```
>>> values = d.values()
>>> values
dict_values([2, 1, 3])
```

values() ফাংশনের মাধ্যমে ডিকশনারির ভ্যালুগুলো ভ্যালু-অবজেক্ট হিসেবে পাওয়া যায়।

```
>>> len(keys)
3
len() ফাংশন দিয়ে কি-অবজেক্টের মোট সংখ্যা দেখা সম্ভব।
```

```
>>> del d['one']
>>> keys
dict_keys(['two', 'three'])
```

ডিকশনারি থেকে কি ডিলিট করলে কি-অবজেক্ট এবং ভ্যালু-অবজেক্টও পরিবর্তন হয়ে যাবে।

```
>>> values
dict_values([2, 3])
>>> n = 0
>>> for val in values:
n += val
>>> n
5
```

এখানে আমরা ফর লুপ চালিয়ে ভ্যালু-অবজেক্টের উপাদানগুলোর যোগফল বের করেছি।

এছাড়া টিউপল এবং ডিকশনারির আরও ফাংশন আছে। এ সম্পর্কে বিস্তারিত জানতে পাইথনের ডকুমেন্ট দেখা যেতে পারে [\[১\]](#)

ফিডব্যাক : ahmadalsajid@gmail.com