

S.P Group of Institute



Programming and Problem Solving Through Python Language Notes

Unit 5: Sequence Data Types

Lists, Tuples और Dictionary



- ★ **List** एक ordered और mutable collection है।
- ★ **Tuple** ordered लेकिन immutable collection है।
- ★ **Dictionary** key-value pairs में data रखता है।

Syntax

```
my_list = [ ]  
my_tuple = ( )  
my_dict = { key: value }
```

Examples

```
my_list = [10, 20, 30]           # List example  
my_tuple = (10, 20, 30)         # Tuple example  
student = { "name": "Ravi", "marks": 85 }  
                                # Dictionary example
```



विशेषता	List	Tuple	Dictionary
ordered	हाँ, ordered है	हाँ, ordered है	Python 3.7+ में ordered है
mutable	हाँ, mutable है	नहीं, immutable है	हाँ, mutable है
access	index द्वारा (0, 1, 2, ...)	index द्वारा (0, 1, 2, ...)	key द्वारा (unique keys)
example	my_list = [10, 20, 30]	my_tuple = (10, 20, 30)	student = { "name": "Ravi", "marks": 85 }



S.P Group of Institute

Programming and Problem Solving Through Python Language Notes

Unit 5: Sequence Data Types

Sequence Operations: Indexing, Slicing, Concatenation और अन्य Operations



Indexing

- Indexing से एक element प्राप्त होता है; positive index left से और negative index right से शुरू होता है।



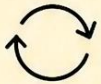
Slicing

- Slicing में [start:stop:step] का उपयोग होता है और stop शामिल नहीं होता।



Concatenation

- Concatenation दो sequences को + से जोड़ता है।



अन्य Operations

- Sequence पर अन्य कई Operations उपलब्ध हैं, जैसे len(), in, not in, count(), index() आदि।

उदाहरण:

```
nums = [10, 20, 30, 40, 50]
nums[0] = 10
nums[-1] = 50
```

उदाहरण:

```
nums = [10, 20, 30, 40, 50]
nums[1:4] = [20, 30, 40]
nums[::2] = [10, 30, 50]
```

उदाहरण:

```
[1, 2] + [3, 4] = [1, 2, 3, 4]
```

अन्य Operations

Operation	विवरण	उदाहरण (nums = [10, 20, 30, 40, 50])
len()	sequence में elements की संख्या वापस करता है।	len(nums) = 5
in	जाँचता है कि element sequence में मौजूद है या नहीं। (True/False)	20 in nums → True 60 in nums → False
not in	जाँचता है कि element sequence में मौजूद नहीं है या नहीं। (True/False)	60 not in nums → True 20 not in nums → False
count()	दिए गए element की occurrences की संख्या लौटाता है।	nums.count(20) = 1 nums.count(60) = 0
index()	दिए गए element का पहला (first) index लौटाता है। (नहीं मिलने पर ValueError देता है।)	nums.index(30) = 2 nums.index(60) → ValueError

S.P Group of Institute

Programming and Problem Solving Through Python Language Notes

Unit 5: Sequence Data Types

Mutability का Concept

- Mutable object को creation के बाद बदला जा सकता है।
- Immutable object को creation के बाद बदला नहीं जा सकता।

Mutable	Immutable
✓ बदला जा सकता है (modify कर सकते हैं)।	✗ बदला नहीं जा सकता (modify नहीं कर सकते)।
✓ Memory में उसी object पर बदलाव होता है।	✗ किसी भी बदलाव पर नया object बनता है।
✓ Example: List, Dictionary	✗ Example: Tuple, str, int, float
✓ Methods जैसे append(), remove(), update() आदि से modify होता है।	✗ Value एक बार set होने के बाद स्थिर (fixed) रहती है।

Mutable Example (List)

```
a = [1, 2, 3]
a[0] = 99
print(a)
# Output: [99, 2, 3]
```

Immutable Example (Tuple)

```
t = (1, 2, 3)
t[0] = 99
print(t)
# Output: TypeError:
'tuple' object does not support
item assignment
```

➤ Mutable Types: List, Dictionary

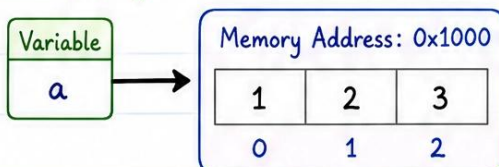


➤ Immutable Types: Tuple, str (string), int (integer), float (floating point)

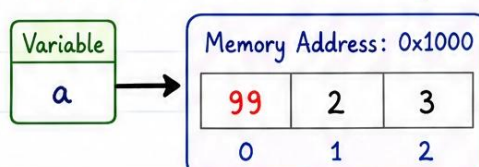


- Memory में क्या होता है? (List का उदाहरण)

(1) शुरुआत में



(2) $a[0] = 99$ के बाद (Same object बदला)



List में same object (same memory address) बदला जाता है।

निष्कर्ष: List में element बदलने पर वही object (same memory) modify होता है।



ध्यान दें: variable को नया object assign करना अलग बात है।

S.P Group of Institute



Programming and Problem Solving Through Python Language Notes



Unit 5: Sequence Data Types

Examples: Maximum, Minimum, Mean और Linear Search

Sequence (जैसे List या Tuple) पर कई उपयोगी ऑपरेशन्स किए जा सकते हैं।
यहाँ हम Maximum, Minimum, Mean और Linear Search के उदाहरण देखेंगे।

①

Maximum और Minimum

List में उपस्थित सबसे बड़ा (maximum) और सबसे छोटा (minimum) element निकालने के लिए `max()` और `min()` का उपयोग करते हैं।

```
nums = [12, 5, 27, 9, 18]
print(max(nums))
print(min(nums))
```

आउटपुट:

27

→ `print(max(nums))` का परिणाम 27

5

→ `print(min(nums))` का परिणाम 5

②

Mean

Mean (औसत) निकालने के लिए सभी elements का योग लेकर कुल elements की संख्या से भाग करते हैं।

```
nums = [12, 5, 27, 9, 18]
mean = sum(nums) / len(nums)
print(mean)
```

आउटपुट:

14.2

→ `sum(nums) = 71`

→ `len(nums) = 5`

→ `mean = 71 / 5 = 14.2`

③

Linear Search

Linear Search में हम सूची के पहले element से शुरू करके एक-एक करके तुलना करते हैं जब तक कि target element मिल न जाए या सूची समाप्त न हो जाए।

```
nums = [12, 5, 27, 9, 18]
target = 27
found = False
for x in nums:
    if x == target:
        found = True
        break
print(found)
```

आउटपुट:

True

→ target (27) सूची में मौजूद है,
इसलिए परिणाम True आया।

Linear Search की प्रक्रिया (Step-by-Step):

पहले element
से शुरू



एक-एक करके
तुलना



match मिले
तो stop



नहीं मिले तो
not found

* उपर्युक्त सभी ऑपरेशन्स (Maximum, Minimum, Mean और Linear Search) Tuple पर भी बिल्कुल इसी प्रकार काम करते हैं।



Programming and Problem Solving Through Python Language Notes

Unit 5: Sequence Data Types

List में Elements की Frequency गिनना using Dictionary



Frequency का अर्थ है किसी element के आने की संख्या.

Dictionary में element को key और उसकी count को value रखते हैं.

```
</>
```

```
nums = [2, 3, 2, 5, 3, 2]
```

```
freq = {}
```

```
for x in nums:
```

```
    if x in freq:
```

```
        freq[x] += 1
```

```
    else:
```

```
        freq[x] = 1
```

```
print(freq)
```



Trace (Step-by-Step)

Element	Dictionary after update
2	{2: 1}
3	{2: 1, 3: 1}
2	{2: 2, 3: 1}
5	{2: 2, 3: 1, 5: 1}
3	{2: 2, 3: 2, 5: 1}
2	{2: 3, 3: 2, 5: 1}



Output:

```
{2: 3, 3: 2, 5: 1}
```



Takeaway:

यह तरीका किसी भी List के repeated elements की counting के लिए उपयोगी है।

