

Class #05: Python's Built-in Data Structures

- ❖ Working with lists and performing operations like indexing, slicing, and negative indexing.
- ❖ Introduction to tuples and sets in Python.
- ❖ Exploring dictionaries and how to use them.
- ❖ Assignment #5: Practice with data structures (lists, tuples, sets, dictionaries).

Python's Built-in Data Structures হলো Python প্রোগ্রামিং ভাষায় প্রাক-নির্ধারিত ডেটা স্ট্রাকচার, যা ডেটা সংরক্ষণ এবং পরিচালনা করার জন্য সরাসরি ব্যবহার করা যায়। এগুলি খুবই কার্যকর এবং ব্যবহারে সহজ, কারণ এগুলি Python ভাষায় অন্তর্নির্মিত (পাইথন বানানোর সময় থেকেই ছিল) এবং সেগুলোর জন্য আলাদা লাইব্রেরি বা মডিউল ব্যবহার করার প্রয়োজন হয় না।

There are four collection data types in Python:

1. List: A collection that is **ordered** and **changeable**. It allows **duplicate members**.
2. Tuple: A collection that is **ordered** and **unchangeable**. It allows **duplicate members**.
3. Set: A collection that is **unordered**, **unchangeable** (but can be modified), and **unindexed**. It does not allow **duplicate members**.
4. Dictionary: A collection that is **ordered** and **changeable**. It does not allow **duplicate members**.

Lists, tuples, sets, dict এর মধ্যে একই সাথে int, float, complex, str, bool ডেটা থাকতে পারে। আগে তো আমরা শুধুমাত্র দেখেছিলাম একটা চলক/variable এর মধ্যে শুধুমাত্র ১টা ডেটা থাকতে পারে। এবার lists, tuples, sets, dict এর মধ্যে আমরা দেখবো, একগুচ্ছ ডেটা থাকে। Lists, tuples, sets, dict হলো সিকুয়েন্স ডেটা।

Data Type	Ordered/ ক্রমিক নং থাকে	Changeable/ মান পরিবর্তন করা যায়	Allows Duplicate Members/ একই মান দুইবার থাকতে পারে ?	Indexed/ ক্রমিক নং থাকে
List	Yes	Yes	Yes []	Yes
Tuple	Yes	No	Yes ()	Yes
Set	No	Yes	No {}	No
Dictionary	Yes	Yes	No {}	Yes

- ❖ Working with lists and performing operations like indexing, slicing, and negative indexing.

List: A collection that is **ordered** and **changeable(mutable)**. It allows **duplicate members**.

List (লিস্ট) একটি ডেটা স্ট্রাকচার যা পাইটনে ব্যবহৃত হয় এবং এটি অর্ডারড (ordered) এবং পরিবর্তনযোগ্য (mutable) একটি সংগ্রহ বা সিকোয়েন্স ডেটা টাইপ। লিস্টের মধ্যে একাধিক মান সংরক্ষণ করা যায় এবং এই মানগুলোতে পরিবর্তন, যোগ বা মুছে ফেলা সম্ভব।

List এর বৈশিষ্ট্য:

1. **অর্ডারড (Ordered):** লিস্টে যে কোনো আইটেম নির্দিষ্ট এক অর্ডারে থাকে, অর্থাৎ তাদের অবস্থান বা ইনডেক্স থাকবে।
2. **পরিবর্তনযোগ্য (Mutable):** একবার একটি লিস্ট তৈরি হলে, আপনি তার মান পরিবর্তন, যোগ বা মুছে ফেলতে পারেন।
3. **ডুপ্লিকেট মান সমর্থন করে:** লিস্টে একই মান একাধিকবার থাকতে পারে।
4. **হিসাবযোগ্য (Indexed):** লিস্টের সব আইটেমের একটি নির্দিষ্ট ইনডেক্স থাকে, যেখানে প্রথম আইটেমের ইনডেক্স 0, দ্বিতীয়টির 1, ইত্যাদি।

আমাকে বলা হলো, একটা list বানাতে, যারা যারা python শিখতে চায়। আমি সবার নাম দিয়ে একটা লিস্ট বানালাম। তোমরা এই কোড pycharm এ অবশ্যই চালু করবে।

```
# লিস্টের মধ্যে নামগুলোর একটি সংগ্রহ
names = ["Fahim", "Tasin", "Fardin", "Farzana", "Worthy", "Limon", "Athai",
          "Yeaj Uddin", "Ratul", "Nabila", "Nusrath", "Nazrul", "Siam",
          "Sarowar", "Jobayer", "Shamol", "Ankur", "Tanvir", "Tarif", "Tanha",
          "Mahamudul"]

# লিস্টে থাকা সকল নাম প্রিন্ট করা হবে
print(f"Print all names: {names}")

# লিস্টে মোট সংখ্যক নাম প্রিন্ট করা হবে
print(f"Number of total people: {len(names)}")

# লিস্টের ডাটা টাইপ প্রিন্ট করা হবে
print(f"Data type of 'names': {type(names)}")

Print all names: ['Fahim', 'Tasin', 'Fardin', 'Farzana', 'Worthy', 'Limon', 'Athai', 'Yeaj Uddin',
'Ratul', 'Nabila', 'Nusrath', 'Nazrul', 'Siam', 'Sarowar', 'Jobayer', 'Shamol', 'Ankur', 'Tanvir',
'Tarif', 'Tanha', 'Mahamudul']

Number of total people: 21
```

Data type of 'names': <class 'list'>

পাইথনে লিস্ট বানানোর জন্য আপনার লিস্টের আইটেমগুলোকে স্কয়ার ব্র্যাকেট [] এর মধ্যে রাখেন, এবং আইটেমগুলোকে কমা দিয়ে আলাদা করেন। ব্যস তৈরি হয়ে গেলো লিস্ট।

Python - Access List Items: access মানে সোজা বাংলায় output এ দেখা।

আমরা list এর একক মান বা গুচ্ছ মান মান দেখতে পারি। slicing আমরা str এ পড়েছিলাম।

https://www.w3schools.com/python/python_lists_access.asp

Creating a list of names

```
names = ["Fahim", "Tasin", "Fardin", "Farzana", "Worthy", "Limon", "Athai",
        "Yeaj Uddin", "Ratul", "Nabila", "Nusrath", "Nazrul", "Siam",
        "Sarowar", "Jobayer", "Shamol", "Ankur", "Tanvir", "Tarif", "Tanza",
        "Mahamudul"]
```

Accessing specific items from the list

```
print(f"names[0]: {names[0]}")
print(f"names[10]: {names[10]}")
print(f"names[-1]: {names[-1]}")
```

List slicing - getting a specific part of the list

```
print(f"names[2:5]: {names[2:5]}")
print(f"names[:5]: {names[:5]}")
print(f"names[11:]: {names[11:]}")
print(f"names[-2: -5]: {names[-5: -2]}")
```

names[0]: Fahim

names[10]: Nusrath

names[-1]: Mahamudul

names[2:5]: ['Fardin', 'Farzana', 'Worthy']

names[:5]: ['Fahim', 'Tasin', 'Fardin', 'Farzana', 'Worthy']

names[11:]: ['Nazrul', 'Siam', 'Sarowar', 'Jobayer', 'Shamol', 'Ankur', 'Tanvir', 'Tarif', 'Tanza',
'Mahamudul']

names[-2: -5]: ['Ankur', 'Tanvir', 'Tarif']

List: A collection that is **ordered** and **changeable(mutable)**. It allows **duplicate members**.

যদি আমরা খেয়াল করি, names[0]: Fahim, names[10]: Nusrath, names[-1]: Mahamudul এইগুলো সংখ্যা অনুযায়ী সাজানো আছে। Order মানে হলো, সাজানো। তার মানে, list এ মানগুলোর ক্রমিক বা order থাকে।

Python Change list value: https://www.w3schools.com/python/python_lists_change.asp

```
# List of names
names = ["Fahim", "Tasin", "Fardin", "Farzana", "Worthy", "Limon", "Athai",
         "Yeaj Uddin", "Ratul", "Nabila", "Nusrath", "Nazrul", "Siam",
         "Sarowar", "Jobayer", "Shamol", "Ankur", "Tanvir", "Tarif", "Tanha",
         "Mahamudul"]

# Change the second item (index 1) of the list to "MINHAZUL"
names[1] = "MINHAZUL"
print(f"Updated list after changing second item: {names}")

# Replace a range of values (index 2 to 17) with a single value "Kabir"
names[2:18] = ["Kabir"]
print(f"List after replacing range with 'Kabir': {names}")

# List of fruits
thislist = ["apple", "banana", "cherry"]

# Replace the second item (index 1) with multiple values "watermelon", "mango", and
# "lemon"
thislist[1:2] = ["watermelon", "mango", "lemon"]
print(f"List after inserting multiple fruits: {thislist}")

# Insert a single value "chicken" at position 2 (index 2) in the list
thislist.insert(2, "chicken")
print(f"List after inserting 'chicken' at position 2: {thislist}")
```

Updated list after changing second item: ['Fahim', 'MINHAZUL', 'Fardin', 'Farzana', 'Worthy', 'Limon', 'Athai', 'Yeaj Uddin', 'Ratul', 'Nabila', 'Nusrath', 'Nazrul', 'Siam', 'Sarowar', 'Jobayer', 'Shamol', 'Ankur', 'Tanvir', 'Tarif', 'Tanha', 'Mahamudul']

List after replacing range with 'Kabir': ['Fahim', 'MINHAZUL', 'Kabir', 'Tarif', 'Tanha', 'Mahamudul']

List after inserting multiple fruits: ['apple', 'watermelon', 'mango', 'lemon', 'cherry']

List after inserting 'chicken' at position 2: ['apple', 'watermelon', 'chicken', 'mango', 'lemon', 'cherry']

List: A collection that is **ordered** and **changeable(mutable)**. It allows **duplicate members**.

আমরা যদি খেয়াল করি, list এর মান পরিবর্তন করা যায়, তারমানে, list changeable(mutable).

Python - Add List Items: https://www.w3schools.com/python/python_lists_add.asp

append() Method:

বাংলা অর্থ: সর্বশেষে যুক্ত করা

ব্যবহার: append() method এর সাহায্যে list এর সর্বশেষে একটি মান যুক্ত করা যায়। এটি শুধুমাত্র একটি একক মান যুক্ত করতে ব্যবহৃত হয়।

insert() Method:

বাংলা অর্থ: সন্নিবেশ

ব্যবহার: insert() method এর সাহায্যে list এর নির্দিষ্ট স্থানে একটি মান যুক্ত করা যায়। এটি একটি নির্দিষ্ট index এ মান সন্নিবেশ করে, যেখানে আপনি চান।

extend() Method:

বাংলা অর্থ: প্রসারিত

ব্যবহার: extend() method এর সাহায্যে list এর শেষে একাধিক মান যুক্ত করা যায়। এর মাধ্যমে আপনি tuple, set, dictionary ইত্যাদিও list এ যোগ করতে পারেন।

```
# List of names
names = ["Fahim", "Tasin", "Fardin", "Farzana", "Worthy"]
print(f"Initial names list: {names}")

# Using append() method to add a single item
names.append("Limon")
print(f"After using append(): {names}")

# Using insert() method to add an item at a specific index
names.insert(2, "minhazul kabir")
print(f"After using insert(): {names}")

# Another list of names to extend with
name2 = ["Athai", "Yeaj Uddin", "Ratul", "Nabila", "Nusrath"]

# Using extend() method to add multiple items from another list
names.extend(name2)
print(f"After using extend() with another list: {names}")
```

```
# Tuple of names to extend with
nameTuple = ("Nazrul", "Siam", "Sarowar", "Jobayer", "Shamol", "Ankur", "Tanvir", "Tarif",
"Tanha", "Mahamudul")

# Using extend() method to add items from a tuple
names.extend(nameTuple)
print(f"After using extend() with a tuple: {names}")
```

Initial names list: ['Fahim', 'Tasin', 'Fardin', 'Farzana', 'Worthy']

After using append(): ['Fahim', 'Tasin', 'Fardin', 'Farzana', 'Worthy', 'Limon']

After using insert(): ['Fahim', 'Tasin', 'minhazul kabir', 'Fardin', 'Farzana', 'Worthy', 'Limon']

After using extend() with another list: ['Fahim', 'Tasin', 'minhazul kabir', 'Fardin', 'Farzana', 'Worthy', 'Limon', 'Athai', 'Yeaj Uddin', 'Ratul', 'Nabila', 'Nusrath']

After using extend() with a tuple: ['Fahim', 'Tasin', 'minhazul kabir', 'Fardin', 'Farzana', 'Worthy', 'Limon', 'Athai', 'Yeaj Uddin', 'Ratul', 'Nabila', 'Nusrath', 'Nazrul', 'Siam', 'Sarowar', 'Jobayer', 'Shamol', 'Ankur', 'Tanvir', 'Tarif', 'Tanha', 'Mahamudul']

List: A collection that is **ordered** and **changeable(mutable)**. It allows **duplicate members**.

আমরা যদি খেয়াল করি, list এ মান যুক্ত করা যায়, তারমানে, list **changeable(mutable)**.

নিজে নিজে এইগুলো list দেখো এবং pycharm এ কাজ করাও।

https://www.w3schools.com/python/python_lists_remove.asp

https://www.w3schools.com/python/python_lists_loop.asp

https://www.w3schools.com/python/python_lists_comprehension.asp

https://www.w3schools.com/python/python_lists_sort.asp

https://www.w3schools.com/python/python_lists_copy.asp

https://www.w3schools.com/python/python_lists_join.asp

https://www.w3schools.com/python/python_lists_methods.asp

❖ Introduction to tuples and sets in Python.

Tuple এর বৈশিষ্ট্য:

1. **অপরিবর্তনীয় (Immutable):** একবার টাপলে মান সংরক্ষিত হলে, সেটি পরিবর্তন করা যায় না। মান পরিবর্তন, যোগ বা মুছে ফেলা সম্ভব নয়।

2. ক্রম অনুসরণ করে (Ordered): টাপলে মানগুলোর একটি নির্দিষ্ট অর্ডার থাকে এবং তাদের অবস্থান (index) নির্দিষ্ট থাকে।
3. অনুপ্রেরণা যোগ্য (Iterable): টাপল একটি iterable ডেটা টাইপ, অর্থাৎ, টাপলের মধ্যে থাকা আইটেমগুলোকে এক এক করে ব্যবহার করা যায়।
4. ডুপ্লিকেট মান সমর্থন করে: টাপলে একাধিকবার একই মান থাকতে পারে।

আমাকে বলা হলো, একটা tuple বানাতে, যারা যারা SUST এ picnic করতে চায়। আমি সবার নাম ব্যবহার করে একটা tuple বানালাম। কে কত প্লেট খাবার থাবে, সেটা শোনার পরে রান্না শুরু হলো। রান্না শুরু হবার পরে, যেমন meal থেকে নাম কাটাতে পারবে না। কারণ, অতিরিক্ত খাবার তো ছুট করা রান্না সম্ভব না। আবার, খাবার নষ্ট করাও যাবে না। তোমরা এই কোড pycharm এ অবশ্যই চালু করবে।

Tuple: A collection that is **ordered** and **unchangeable(immutable)**. It allows **duplicate members**.

```
# tuple মধ্যে নামগুলোর একটি সংগ্রহ
names = ("Fahim", "Tasin", "Fardin", "Farzana", "Worthy", "Limon", "Athai",
          "Yeaj Uddin", "Ratul", "Nabila", "Nusrath", "Nazrul", "Siam",
          "Sarowar", "Jobayer", "Shamol", "Ankur", "Tanvir", "Tarif", "Tanha",
          "Mahamudul")
```

```
# tuple থাকা সকল নাম প্রিন্ট করা হবে
print(f"Print all names: {names}")
```

```
# tuple মোট সংখ্যক নাম প্রিন্ট করা হবে
print(f"Number of total people: {len(names)}")
```

```
# tuple ডাটা টাইপ প্রিন্ট করা হবে
print(f"Data type of 'names': {type(names)}")
```

```
Print all names: ('Fahim', 'Tasin', 'Fardin', 'Farzana', 'Worthy', 'Limon', 'Athai', 'Yeaj Uddin',
'Ratul', 'Nabila', 'Nusrath', 'Nazrul', 'Siam', 'Sarowar', 'Jobayer', 'Shamol', 'Ankur', 'Tanvir',
'Tarif', 'Tanha', 'Mahamudul')
```

```
Number of total people: 21
```

```
Data type of 'names': <class 'tuple'>
```

পাইথনে tuple তৈরি করার জন্য আপনার আইটেমগুলোকে Parenthesis ব্যাকেট () এর মধ্যে রাখেন, এবং আইটেমগুলোকে কমা দিয়ে আলাদা করেন। ব্যস তৈরি হয়ে গেলো tuple()।

Python - Access tuple Items: access মানে সোজা বাংলায় output এ দেখা।

আমরা tuple এর একক মান বা গুচ্ছ মান মান দেখতে পারি। slicing আমরা str এ পড়েছিলাম।

https://www.w3schools.com/python/python_tuples_access.asp লিঙ্ক থেকে সবগুলো কোড চালিয়ে দেখতে বলা হলো। tuple এবং list এর ফাংশন প্রায় একই।

Tuple: A collection that is **ordered** and **unchangeable(immutable)**. It allows **duplicate members**.

Order মানে হলো, সাজানো। tuple এ মানগুলোর ক্রমিক বা order অনুযায়ী থাকে।

Python Change tuple value: https://www.w3schools.com/python/python_tuples_update.asp

আমরা জানি, tuple এর মান পরিবর্তন করা যায় না, তারমানে, tuple **unchangeable(immutable)**. tuple এর মান পরিবর্তন করতে হলে তাকে list এ রূপান্তরিত করতে হবে।

```
# Tuple of names
names_tuple = ("Limon", "Athai", "Yeaj Uddin", "Ratul", "Nabila", "Nusrath", "Nazrul",
"Siam", "Sarowar", "Jobayer", "Shamol")

# Convert the tuple to a list to allow modification
names_list = list(names_tuple)
# Change the 3rd item (index 2) in the list
names_list[2] = "MINHAZUL"
# Convert the list back to a tuple
names_tuple = tuple(names_list)
print(f"Updated tuple after changing third item: {names_tuple}")
```

```
# Convert the tuple to a list again for slicing
names_list = list(names_tuple)
# Replace the range of items from index 3 to 9 with the name 'KABIR'
names_list[3:10] = ["KABIR"]
# Convert the list back to a tuple
names_tuple = tuple(names_list)
print(f"Tuple after replacing range with 'KABIR': {names_tuple}")
```

Updated list after changing second item: ('Limon', 'Athai', 'MINHAZUL', 'Ratul', 'Nabila', 'Nusrath', 'Nazrul', 'Siam', 'Sarowar', 'Jobayer', 'Shamol')

List after replacing range with 'Kabir': ('Limon', 'Athai', 'MINHAZUL', 'KABIR', 'Shamol')

যেহেতু tuple এ কোনো মান যুক্ত বা পরিবর্তন করা যায় না, কিন্তু list এ সেটা সম্ভব, তাই যদি tuple এর মান পরিবর্তন করতে হয়, তখন প্রথমে তাকে list এ রূপান্তরিত করতে হয়। এরপর list এ প্রয়োজনীয় পরিবর্তন করে সেটিকে আবার tuple এ রূপান্তরিত করতে হয়।

প্রক্রিয়া:

1. Tuple একটি **immutable (unchangeable)** ডেটা টাইপ। অর্থাৎ একবার তৈরি হওয়ার পর এর মধ্যে থাকা আইটেমগুলো পরিবর্তন করা সম্ভব নয়।
2. কিন্তু list একটি **mutable (changeable)** ডেটা টাইপ। তাই, tuple এর মান পরিবর্তন করতে হলে, প্রথমে তাকে list এ রূপান্তরিত করতে হবে।
3. এরপর, list তে প্রয়োজনীয় পরিবর্তনগুলো যেমন আইটেম যোগ করা, মুছে ফেলা বা পরিবর্তন করা করা যায়।
4. সবশেষে, যখন পরিবর্তন সম্পন্ন হয়ে যাবে, তখন list কে আবার tuple এ রূপান্তরিত করে আগের tuple এর পরিবর্তিত সংস্করণ পাওয়া যাবে।

এভাবে, tuple এর মান পরিবর্তন করা সম্ভব, তবে সেটা করার জন্য তাকে প্রথমে list এ রূপান্তরিত করতে হয় এবং পরবর্তীতে আবার tuple এ রূপান্তরিত করতে হয়।

Tuple: A collection that is **ordered** and **unchangeable(immutable)**. It allows **duplicate members**.

নিজে নিজে এইগুলো tuple দেখো এবং pycharm এ কাজ করাও।

https://www.w3schools.com/python/python_tuples_unpack.asp

https://www.w3schools.com/python/python_tuples_loop.asp

https://www.w3schools.com/python/python_tuples_join.asp

https://www.w3schools.com/python/python_tuples_methods.asp

List এবং Tuple এর মধ্যে প্রধান পার্থক্য হলো তাদের **পরিবর্তনযোগ্যতা** (mutability):

- **List:** এটি একটি **mutable** (পরিবর্তনযোগ্য) ডেটা টাইপ, অর্থাৎ এর মধ্যে থাকা মানগুলোকে পরিবর্তন, যোগ, বা মুছে ফেলা যেতে পারে। উদাহরণস্বরূপ, আপনি একটি তালিকার মধ্যে যেকোনো সময় নতুন মান যোগ করতে পারেন অথবা পুরোনো মান পরিবর্তন করতে পারেন।
- **Tuple:** এটি একটি **immutable** (অপরিবর্তনীয়) ডেটা টাইপ, অর্থাৎ একবার তৈরি হলে এর মধ্যে থাকা মানগুলোকে পরিবর্তন করা যায় না। Tuple এর মান পরিবর্তন করতে হলে, প্রথমে তাকে list এ রূপান্তরিত করতে হবে এবং তারপর সেটি পরিবর্তন করে আবার tuple এ রূপান্তরিত করতে হবে।

উদাহরণ দিয়ে বোঝানো:

ধরা যাক, HSC এর পরে তুমি বিশ্ববিদ্যালয়ে ভর্তি হওয়ার আগে যে বিষয়ে পড়াশোনা করবে তা নিশ্চিত নয়, ঠিক যেমন একটি list এর মান পরিবর্তন করা যায়। কিন্তু, একবার তুমি বিশ্ববিদ্যালয়ে ভর্তি হওয়ার পরে যে বিষয়ে পড়াশোনা শুরু করবে, সেটা পরিবর্তন করা সম্ভব নয়, ঠিক যেমন একটি tuple এর মান পরিবর্তন করা যায় না। যদি তুমি বিশ্ববিদ্যালয়ে পড়ার বিষয় পরিবর্তন করতে চাও, তাহলে তোমাকে আবার HSC (যেমন, list এর মধ্যে থাকা মান

পরিবর্তন করতে হলে প্রথমে তা list এ রূপান্তর করতে হবে) অবস্থায় ফিরে যেতে হবে, অর্থাৎ, প্রথমে tuple কে list এ রূপান্তরিত করতে হবে এবং তারপর পরিবর্তন করে আবার তা tuple এ রূপান্তরিত করতে হবে।

এভাবে, list পরিবর্তনযোগ্য হলেও tuple পরিবর্তনযোগ্য নয়, তবে tuple এর মান পরিবর্তন করতে চাইলে তা list এ রূপান্তরিত করে পরিবর্তন করতে হবে।

Tuple: A collection that is **ordered** and **unchangeable(immutable)**. It allows **duplicate members**.

set:

Set: A collection that is **unordered**, **unchangeable** (but can be modified), and **unindexed**. It does not allow **duplicate members**.

Set একটি **unordered** (ক্রমবিহীন) এবং **mutable (changeable)** ডেটা টাইপ, যা একক মানের একটি সংগ্রহ। সেটে একই মান (duplicate) থাকতে পারে না এবং এটি **unordered** অর্থাৎ সেটের মধ্যে আইটেমগুলোর কোনো নির্দিষ্ট অর্ডার থাকে না।

Set এর বৈশিষ্ট্য:

1. **Unordered:** সেটের মধ্যে আইটেমগুলোর কোনো নির্দিষ্ট অর্ডার থাকে না।
2. **Unique Elements:** সেটে একে অপরের মতো আইটেম থাকতে পারে না, অর্থাৎ পুনরাবৃত্তি মানের কোন স্থান নেই।
3. **Mutable:** সেটে আইটেম যোগ করা বা মুছে ফেলা যেতে পারে।
4. **No Indexing:** যেহেতু সেটের আইটেমগুলো নির্দিষ্ট অর্ডারে থাকে না, তাই সেটের আইটেমগুলোতে সরাসরি অ্যাক্সেস করা যায় না।

ব্যবহার:

- সেট ব্যবহৃত হয় যখন আপনি চান যে একাধিক মানের মধ্যে পুনরাবৃত্তি না হয় এবং অর্ডারের প্রতি মনোযোগ না দিয়ে শুধুমাত্র নির্দিষ্ট আইটেমগুলো হালনাগাদ বা মুছে ফেলতে।

আমরা ছোট বেলায় গণিতে যে set পড়েছিলাম, সেটা এটাই।

আমাকে বলা হলো, একটা set বানাতে যারা পিকনিকে যাবে। আমাকে Limon, Tasin, Ankur এই 3 জনে 3 sheet দিলো। আমি তাদের sheet এ দেখলাম, একই নামের পুনরাবৃত্তি। তোমরা এই কোড pycharm এ অবশ্যই চালু করবে।

```
# Set of names
picnic_names = {"Fahim", "Tasin", "Tanvir", "Athai", "Fardin", "Fahim", "Tanvir", "Athai"}

# Print all unique names
print(f"Print all unique names: {picnic_names}")

Print all unique names: {'Fardin', 'Tanvir', 'Tasin', 'Fahim', 'Athai'}
```

পাইথনে set তৈরি করার জন্য আপনাকে সেটের আইটেমগুলোকে কার্লি ব্র্যাকেট { } এর মধ্যে রাখতে হবে, এবং আইটেমগুলোকে কমা দিয়ে আলাদা করতে হবে। ব্যস, তৈরি হয়ে গেলো set।

এইখানের, output গুলো যা হলুদ কালি দিয়ে মার্ক করা হয়েছে, তাদের ক্রম পরিবর্তনশীল। মানে, তুমি যদি python code চালু করার পরে এই নাম গুলো অন্য ক্রমে আসবে।

Set: A collection that is **unordered**, **unchangeable** (but can be modified), and **unindexed**. It does not allow **duplicate members**.

Set এর মাধ্যমে duplicate ডেটা remove করা হয়, যা মেশিন লার্নিং-এ বিশেষভাবে দরকার। কারণ, মেশিন লার্নিং-এ একাধিক পুনরাবৃত্তি মান (duplicate values) থাকলে তা ফলস বা অপ্রয়োজনীয় তথ্যের সৃষ্টি করতে পারে। Set এ শুধুমাত্র একক মান থাকতে পারে, তাই এতে duplicate values স্বয়ংক্রিয়ভাবে সরিয়ে ফেলা হয়।

Set is Unchangeable: এর অর্থ হলো, set এর value বা মান replace করা যায় না, অর্থাৎ,

একবার যে মান সেটে যোগ করা হয়েছে, তা সরাসরি replace করা সম্ভব নয়। তবে, set এ নতুন মান যোগ (add) করা বা পুরানো মান মুছে ফেলা (remove) যায়।

উদাহরণস্বরূপ: আমি যদি JU (জাহাঙ্গীরনগর বিশ্ববিদ্যালয়) এ চান্দপাণ্ড হয়ে একটি নির্দিষ্ট সিট পেয়ে থাকি, তবে আমি সেই সিট অন্য কাউকে অনুদান বা বদল করে দিতে পারব না। এটি যেমন সম্ভব নয়, তেমনি set এর মানও edit করা যায় না। তবে, যদি আমি চলে যাই, তাহলে আমার সিটটি মুছে (remove) ফেলা হবে এবং যদি নতুন কোনো শিক্ষার্থী আসেন, তবে সেটি যোগ (add) করা যাবে।

এভাবে, set এর মান replaceable নয়, তবে সেটে নতুন মান যোগ বা পুরানো মান মুছে ফেলা সম্ভব।

নিজে নিজে এইগুলো set দেখো এবং pycharm এ কাজ করাও।

https://www.w3schools.com/python/python_sets.asp

https://www.w3schools.com/python/python_sets_access.asp

https://www.w3schools.com/python/python_sets_add.asp

https://www.w3schools.com/python/python_sets_remove.asp

https://www.w3schools.com/python/python_sets_loop.asp

https://www.w3schools.com/python/python_sets_join.asp

https://www.w3schools.com/python/python_sets_methods.asp

পাইথনে set এবং tuple এর মান সরাসরি replace করা সম্ভব নয়, কারণ এগুলো immutable (unchangeable) ডেটা স্ট্রাকচার। তবে, আপনি type casting ক্ষেত্রে এগুলোকে list এ রূপান্তরিত করে মান পরিবর্তন করতে পারেন, এবং পরে সেগুলোকে আবার set বা tuple তে রূপান্তরিত করতে পারেন।

প্রক্রিয়া:

1. Set বা Tuple এর মান পরিবর্তন করতে হলে প্রথমে তাদের list এ রূপান্তরিত করতে হবে।
2. এরপর list এ প্রয়োজনীয় পরিবর্তন করা যাবে, কারণ list একটি mutable (changeable) ডেটা স্ট্রাকচার।
3. পরিবর্তিত list কে আবার set বা tuple তে রূপান্তরিত করা যাবে।

এইভাবে, type casting এর মাধ্যমে set বা tuple এর মান পরিবর্তন করা সম্ভব হয়।

C, C++, Java	Python
Array একটি নির্দিষ্ট ধরনের ডেটা ধারণ করে (যেমন int, float, ইত্যাদি)	Array হিসেবে Python এ বিভিন্ন ডেটা স্ট্রাকচার রয়েছে (যেমন List, Tuple, Set, Dict)
Array তে শুধুমাত্র একটি ধরণের ডেটা থাকতে পারে (যেমন সব int বা সব float)	Python এ List, Tuple ইত্যাদিতে একাধিক ধরণের ডেটা থাকতে পারে (int, str, float, bool, ইত্যাদি)
Mostly Array তে int টাইপের ডেটা থাকে	Python এর List তে সাধারণত str টাইপের ডেটা থাকে (যেমন ['apple', 'banana', 'cherry'])
C, C++, Java তে Array এক ধরনের কন্টেইনার যা শুধুমাত্র একই ধরনের ডেটা ধারণ করে	Python এ List, Tuple, Set, Dict একাধিক ডেটা টাইপ ধারণ করতে পারে, এবং এগুলোর মধ্যে বিভিন্ন বৈশিষ্ট্য থাকে (অর্ডার, ইউনিক আইটেম ইত্যাদি)
সফটওয়্যার বা অ্যাপ্লিকেশন তৈরি করার জন্য, সাধারণত সংখ্যাসমূহ (যেমন int) ব্যবহার করা হয়, কারণ সফটওয়্যার ডেভেলপমেন্টে গাণিতিক হিসাব বা তথ্য প্রক্রিয়াকরণের জন্য সংখ্যার প্রয়োজন পড়ে।	অন্যদিকে, গবেষণামূলক কাজে এবং ডেটা বিশ্লেষণে, সাধারণত স্ট্রিং বা আক্ষরিক ধরনের ডেটা (যেমন str) ব্যবহার করা হয়, কারণ গবেষণায় প্রায়ই টেক্সট ডেটা, বর্ণনা, বা অন্যান্য আক্ষরিক তথ্যের প্রয়োজন হয়।

পাইথনে আপনি যদি শুধুমাত্র সংখ্যার জন্য আরো কার্যকরী array চান, তবে Python এ NumPy লাইব্রেরি ব্যবহার করা হয়। NumPy ব্যবহার করে আপনি python এ Array তৈরি করতে পারেন যেখানে শুধু সংখ্যার ডেটা থাকবে এবং এটি আরো দ্রুতগতিতে গণনা করতে সাহায্য করবে।

তোমরা কি জানো, লুপ ব্যবহার করলে list, tuple, বা set এর ফরম্যাট ভেঙ্গে যায়? নিচের কোডটি চালু করাও, কি output পেলে ?

```
list_names = ["Fahim", "Tasin", "Fardin", "Farzana", "Worthy", "Limon", "Athai",
    "Yeaj Uddin", "Ratul", "Nabila", "Nusrath", "Nazrul", "Siam",
    "Sarowar", "Jobayer", "Shamol", "Ankur", "Tanvir", "Tarif", "Tanha",
    "Mahamudul"]
print("list:", end=" ")
for x in list_names:
    print(x,end=" ")
```

```

tuple_names=("Fahim", "Tasin", "Fardin", "Farzana", "Worthy", "Limon", "Athai",
            "Yeaj Uddin", "Ratul", "Nabila", "Nusrath", "Nazrul", "Siam",
            "Sarowar", "Jobayer", "Shamol", "Ankur", "Tanvir", "Tarif", "Tanha",
            "Mahamudul")
print("\ntuple:")
for y in tuple_names:
    print(y,end=" ")
set_names={"Fahim", "Tasin", "Fardin", "Farzana", "Worthy", "Limon", "Athai",
           "Yeaj Uddin", "Ratul", "Nabila", "Nusrath", "Nazrul", "Siam",
           "Sarowar", "Jobayer", "Shamol", "Ankur", "Tanvir", "Tarif", "Tanha",
           "Mahamudul"}
print("\nset:")
for z in set_names:
    print(z, end=" ")

```

❖ Exploring dictionaries and how to use them.

Dictionary: A collection that is **ordered** and **changeable**. It does not allow **duplicate members**.

Dict (ডিকশনারি) একটি ডেটা স্ট্রাকচার যা পাইথনে ব্যবহৃত হয় এবং এটি অর্ডারড (ordered) এবং পরিবর্তনযোগ্য (mutable) একটি সংগ্রহ ডেটা টাইপ। ডিকশনারির মধ্যে কী (key) এবং মান (value) এর জোড়া সংরক্ষণ করা হয়, যেখানে প্রতিটি কী অনন্য এবং অপরিবর্তনীয় (immutable) থাকে।

Dict এর বৈশিষ্ট্য:

1. **অর্ডারড (Ordered):** পাইথন 3.7+ সংস্করণে ডিকশনারির উপাদানগুলি ইনসার্চনের অর্ডারে থাকে, অর্থাৎ একটি ডিকশনারির মানগুলো তাদের সংরক্ষিত হওয়ার অবস্থান অনুসারে থাকে।
2. **পরিবর্তনযোগ্য (Mutable):** একবার একটি ডিকশনারি তৈরি হলে, আপনি তার মান পরিবর্তন, যোগ বা মুছে ফেলতে পারেন।
3. **কী-মান জোড়া (Key-Value Pairs):** ডিকশনারির মধ্যে প্রতিটি উপাদান একটি কী (key) এবং তার সংশ্লিষ্ট মান (value) এর জোড়া হিসেবে থাকে।
4. **কী অনন্য (Unique Keys):** ডিকশনারির প্রতিটি কী অনন্য হতে হবে, একাধিক একই কী থাকতে পারে না। তবে মানের মধ্যে ডুপ্লিকেট থাকতে পারে।
5. **হিসাবযোগ্য (Indexed):** ডিকশনারিতে কী এর মাধ্যমে মান অ্যাক্সেস করা হয়, তবে ইনডেক্স ব্যবহার করা হয় না, কেবল কী-ই ব্যবহার করা হয়।

Dictionary: A collection that is **ordered** and **changeable**. It does not allow **duplicate members**.

Bracket (ব্র্যাকেট) বিভিন্ন ধরনের ডেটা স্ট্রাকচার তৈরি করতে ব্যবহৃত হয়। আমরা যখন **square brackets** [], **parentheses** (), বা **curly brackets** { } ব্যবহার করি, তখন তাদের মাধ্যমে বিভিন্ন ডেটা স্ট্রাকচার তৈরি হয়, যা যথক্রমে **list**, **tuple**, এবং **set** এর রূপান্তরিত হয়। কিন্তু **dict** (ডিকশনারি) এর ক্ষেত্রে, আমরা কেবলমাত্র মান (values) নয়, বরং **কী** (key) এবং **মান** (value) এর জোড়া রাখি।

Dict (ডিকশনারি) এর **keys** চলকের মতো, যার মাধ্যমে সংশ্লিষ্ট **value** পাওয়া যায়। প্রতিটি কী একটি নির্দিষ্ট মানের সাথে যুক্ত থাকে এবং এটি ডেটা দ্রুত অনুসন্ধান, পরিবর্তন বা মুছে ফেলার জন্য ব্যবহৃত হয়।

Machine learning এ dict সবচেয়ে বেশি ব্যবহৃত হয়, কারণ এটি key-value র জোড়া সংরক্ষণ করে, যা সহজে এক্সেস করা যায়। মডেল ট্রেনিং বা ভবিষ্যদ্বাণী প্রক্রিয়ায় ফিচার বা ইনপুট ভেরিয়েবলকে কী হিসেবে ব্যবহার করা হয়। অন্যদিকে, list, tuple, এবং set এর তুলনায় dict অধিক কার্যকরী, কারণ এটি দ্রুত ডেটা এক্সেস এবং পরিবর্তন করতে সক্ষম।

আমরা যদি Excel দেখার মাধ্যমে বুঝতে চাই, তাহলে এটি ঠিক হাজিরা খাতার মতো। Excel এর উপরে **heading** বা **শিরোনাম** থাকে, যা সাধারণত প্রিন্ট করা থাকে (দোকান থেকে আসে), আমরা এগুলো লিখি না। এসব **heading** এর সাপেক্ষে আমরা বিভিন্ন **value** বা মান লিখে থাকি। এইভাবে, Excel এ **heading** গুলি বিভিন্ন বিভাগ বা শিরোনাম হিসেবে কাজ করে এবং তার নিচে আমরা তথ্য সংরক্ষণ করি।

Name	Institute	Location	Subject
Ankur	TRU World	Canada	CS
Tasin	BUET	Bangladesh	URP
Nazrul	DU	Bangladesh	STAT
Fardin	MIST	Bangladesh	ME
Tanvir	NJU	China	CSE
Nusrath	SUST	Bangladesh	STAT
Tarif	EWU	Bangladesh	CSE

Excel হচ্ছে একটি 2D array, যেখানে **row** (সারি) এবং **column** (কলাম) থাকে। প্রতিটি **row** এবং **column** এর মাধ্যমে ডেটা সংরক্ষণ করা হয়। যেমন একটি **row** সাধারণত একটি নির্দিষ্ট তথ্যের সারি, এবং একটি **column** একই ধরনের তথ্যের একটি স্তুতি।

প্রথম **row** একটি নির্দিষ্ট ব্যক্তি Ankur তথ্য সারি, এবং **Subject column** শুধুমাত্র subject ধরনের তথ্যের একটি স্তুতি।

"একটি dictionary-তে দুটি উপাদান থাকে: keys() এবং values()। items() মেথডটি এর মধ্যে keys() এবং values() উভয়কেই অন্তর্ভুক্ত করে।"

```

1 # Creating a dictionary with student information
2 student_info = { "Name": "Ankur", "Institute": "TRU World", "Location": "Canada", "Subject": "CS"}
...
3
4 # Printing the keys of the dictionary
5 print(student_info.keys())
6
7 # Printing the values of the dictionary
8 print(student_info.values())
9
10 # Printing the key-value pairs (items) of the dictionary
11 print(student_info.items())
12

```

Run LOOP

```

"C:\Users\Minhazul Kabir\AppData\Local\Programs\Python\Python313\python.exe" D:\PythonProject\MinhazulKabir\.venv\LOOP.py
dict_keys(['Name', 'Institute', 'Location', 'Subject'])
dict_values(['Ankur', 'TRU World', 'Canada', 'CS'])
dict_items([('Name', 'Ankur'), ('Institute', 'TRU World'), ('Location', 'Canada'), ('Subject', 'CS')])
Process finished with exit code 0

```

এখানে:

- keys() মেথডটি dictionary এর সমস্ত keys প্রদান করে।
- values() মেথডটি dictionary এর সমস্ত values প্রদান করে।
- items() মেথডটি key-value জোড়াগুলির তালিকা প্রদান করে, যা keys() এবং values() এর সমন্বয়।

তোমরা কি জানো, লুপ ব্যবহার করলে list, tuple, বা set এর ফরম্যাট ভেঙ্গে যায়? নিচের কোডটি চালু করাও, কি output পেলে ?

```

# student_info dictionary তৈরি করা হচ্ছে
student_info = { "Name": "Ankur", "Institute": "TRU World",
"Location": "Canada", "Subject": "CS"}

# keys() এর ডিফল্ট মান প্রিন্ট করা হচ্ছে
print("Default value (keys) is printed: ")
for x in student_info:
    print(x)

# tuple ফর্ম্যাটে key-value pair প্রিন্ট করা হচ্ছে
print("\nTuple format (key-value pair) is printed: ")
for x in student_info.items():
    print(x)

# key এবং value আলাদা করে প্রিন্ট করা হচ্ছে
print("\nKey and value are printed separately: ")
for x, y in student_info.items():
    print(x, y)

```

Default value (keys) is printed:

Name
Institute
Location
Subject

Tuple format (key-value pair) is printed:

('Name', 'Ankur')
(('Institute', 'TRU World'))
(('Location', 'Canada'))
(('Subject', 'CS'))

Key and value are printed separately:

Name Ankur
Institute TRU World
Location Canada
Subject CS

যেহেতু, শুধু dict তে দুটি অংশ থাকে, সেজন্য x, y ব্যবহার করতে হয়।

একটি dict-তে একটি key এর মধ্যে একাধিক মান থাকতে পারে।



```
student_info = { "Name": ["Nazrul", "Nusrath"], "Institute": ["DU", "SUST"], "Location": ["Bangladesh"], "Subject": ["STAT"] }
print(student_info)
```

```
C:\Users\Minhazul Kabir\AppData\Local\Programs\Python\Python313\python.exe" D:\PythonProject\MinhazulKabir\dict.py
{'Name': ['Nazrul', 'Nusrath'], 'Institute': ['DU', 'SUST'], 'Location': ['Bangladesh'], 'Subject': ['STAT']}
Process finished with exit code 0
```

Name	Institute	Location	Subject
Ankur	TRU World	Canada	CS
Tasin	BUET	Bangladesh	URP
Nazrul	DU	Bangladesh	STAT
Fardin	MIST	Bangladesh	ME
Tanvir	NJU	China	CSE
Nusrath	SUST	Bangladesh	STAT
Tarif	EWU	Bangladesh	CSE

এই হাজিরা খাতা থেকে Nazrul কোথায় পড়াশুনা করে তা বের করতে বলা হলো।

তুমি সর্বপ্রথম দেখবে কোন ঘরে Nazrul আছে। সে ঘর ধরে (সারি অনুযায়ী) ডান পাশে যেতে থাকবে। institute এর ঘরে যা লিখা আছে তা হচ্ছে উত্তর।

```
# Dictionary to store student information with name as the key
student_info = {
    "Ankur": {"Institute": "TRU World", "Location": "Canada", "Subject": "CS"}, # Information about Ankur
    "Tasin": {"Institute": "BUET", "Location": "Bangladesh", "Subject": "URP"}, # Information about Tasin
    "Nazrul": {"Institute": "DU", "Location": "Bangladesh", "Subject": "STAT"}, # Information about Nazrul
    "Fardin": {"Institute": "MIST", "Location": "Bangladesh", "Subject": "ME"}, # Information about Fardin
    "Tanvir": {"Institute": "NJU", "Location": "China", "Subject": "CSE"}, # Information about Tanvir
    "Nusrath": {"Institute": "SUST", "Location": "Bangladesh", "Subject": "STAT"}, # Information about Nusrath
    "Tarif": {"Institute": "EWU", "Location": "Bangladesh", "Subject": "CSE"} # Information about Tarif
}

# Print the name of the institute where Nazrul is studying
print(student_info["Nazrul"]["Institute"]) # Expected output:
# "DU"
```

DU

এই কোড pycharm এ চালু করো। কি পেলে আমাকে whats app এ বাধ্যতামূলক পাঠাবে copy paste করে।

```
student_info = {
    "Ankur": {"Institute": "TRU World", "Location": "Canada", "Subject": "CS"}, # Information about Ankur
    "Tasin": {"Institute": "BUET", "Location": "Bangladesh", "Subject": "URP"}, # Information about Tasin
    "Nazrul": {"Institute": "DU", "Location": "Bangladesh", "Subject": "STAT"}, # Information about Nazrul
    "Fardin": {"Institute": "MIST", "Location": "Bangladesh", "Subject": "ME"}, # Information about Fardin
    "Tanvir": {"Institute": "NJU", "Location": "China", "Subject": "CSE"}, # Information about Tanvir
    "Nusrath": {"Institute": "SUST", "Location": "Bangladesh", "Subject": "STAT"}, # Information about Nusrath
    "Tarif": {"Institute": "EWU", "Location": "Bangladesh", "Subject": "CSE"} # Information about Tarif
}
```

```
# Loop to print key-value pairs
for name, details in student_info.items():
    print(f"Name: {name}")
    for key, value in details.items():
        print(f" {key}: {value}")
    print()
```

এই কোড pycharm এ চালু করো । কি পেলে আমাকে whats app এ বাধ্যতামূলক পাঠাবে copy paste করে ।

```
myDictionary = {
    "listType": ['Apple', 'Cherry', 'Mango', "Apple"],    # listType is a list
    "tupleType": ("Sweet", "Sour", "No Test", "No Test"), # tupleType is a tuple
    "setType": {100, 150, 200, 200}                         # setType is a set
}

# Iterate through each key-value pair in the dictionary
for key, value in myDictionary.items():
    print(key, value) # key is the dictionary key and value is the corresponding value
```

নিজে নিজে এইগুলা dict দেখো এবং pycharm এ কাজ করাও ।

https://www.w3schools.com/python/python_dictionaries.asp

https://www.w3schools.com/python/python_dictionaries_access.asp

https://www.w3schools.com/python/python_dictionaries_change.asp

https://www.w3schools.com/python/python_dictionaries_add.asp

https://www.w3schools.com/python/python_dictionaries_remove.asp

https://www.w3schools.com/python/python_dictionaries_loop.asp

https://www.w3schools.com/python/python_dictionaries_copy.asp

https://www.w3schools.com/python/python_dictionaries_nested.asp

https://www.w3schools.com/python/python_dictionaries_methods.asp

পরিশেষে, List এবং Tuple সঠিকভাবে সিকোয়েন্স বলা যায়, কারণ তাদের উপাদানগুলির একটি নির্দিষ্ট অর্ডার থাকে, তবে Set এবং Dict কেবল কিছু ক্ষেত্রে sequence এর মতো আচরণ করতে পারে, তবে তারা প্রকৃতপক্ষে sequence নয় ।