Python: Getting Started

# Simple Calculations

Python is not only powerful for advanced programming, but it's also great for simple math operations.

## Addition and Subtraction

Add numbers together like this:

1 + 1

2

You don’t need the = symbol to get a result, just write the expression.

Subtract numbers like this:

5 - 3

2

# Working with Lists

In Python, a list is a collection of data stored in one variable. Lists are ordered, mutable (changeable), and can contain duplicates. Items in a list can be of any data type — numbers, strings, or even other lists.

## Creating Lists

Lists are written using square brackets [].  
Here’s an example showing the prices of houses in US dollars:

price\_usd = [97919.38, 300511.20, 293758.14]  
print(price\_usd)

[97919.38, 300511.2, 293758.14]

## Accessing List Items by Index

In Python, indexing starts from 0. So the first item in a list is at index 0, the second at index 1, and so on.  
To access the second item in our price\_usd list:

print(price\_usd[1])

300511.2

## Updating List Items

You can change the value of a list item by accessing it using its index and assigning a new value.

price\_usd[0] = 100000.00  
print(price\_usd)

## Adding and Removing Items

To add an item to the list:

price\_usd.append(450000.00)  
print(price\_usd)

To remove an item from the list:

price\_usd.remove(293758.14)  
print(price\_usd)

## Looping through a List

To loop through and print each item in the list:

for price in price\_usd:  
 print(price)

# References & Further Reading

* A guide to basic math operations in Python
* Python documentation on built-in data types
* Summary of Python data types
* Tutorial on type conversion in Python
* A description of how dictionaries work in Python
* An introduction to JSON
* An introduction to lists in Python
* How to zip lists
* Calculating mean, median, and mode in Python
* A brief tutorial of For Loops

Pandas : Notes

# What is Pandas?

**1. What is Pandas?**

Pandas is a **Python library** for working with **structured data**. It helps you:

* Load data from files (like .csv)
* Organize it in rows and columns (like Excel)
* Analyze, clean, and manipulate the data easily

You first have to import it:

python

CopyEdit

import pandas as pd

**2. What is a DataFrame?**

A **DataFrame** is like a table with rows and columns — the main data structure in Pandas.  
Example:

| **letter** | **number** | **location** |
| --- | --- | --- |
| a | 3 | east |
| b | 2 | east |

**3. How to Read a CSV File**

CSV stands for **Comma Separated Values** — like an Excel file in plain text.

python

CopyEdit

df = pd.read\_csv("filename.csv")

Then you can view the first rows with:

python

CopyEdit

df.head()

**4. Set Index and Reset Index**

**Set Index**: You change the row labels to something more meaningful (like letters or names)  
**Reset Index**: You bring back the default row numbers (0, 1, 2...)

**Practice Example (Fully Explained)**

python

CopyEdit

import pandas as pd

# Your dataset in dictionary format

data = {

"letter": ["a", "b", "c", "d"],

"number": [3, 2, 1, 0],

"location": ["east", "east", "east", "west"],

}

# Convert dictionary to DataFrame

df = pd.DataFrame.from\_dict(data)

# Show original DataFrame

print("Original DataFrame:")

print(df)

**Output:**

css

CopyEdit

letter number location

0 a 3 east

1 b 2 east

2 c 1 east

3 d 0 west

**a) Setting the index to the “letter” column**

python

CopyEdit

df.set\_index('letter', inplace=True)

print("After setting index to 'letter':")

print(df)

**Output:**

css

CopyEdit

number location

letter

a 3 east

b 2 east

c 1 east

d 0 west

**b) Resetting the index**

python

CopyEdit

df.reset\_index(inplace=True)

print("After resetting the index:")

print(df)

**Output:**

css

CopyEdit

letter number location

0 a 3 east

1 b 2 east

2 c 1 east

3 d 0 west

✅ Resetting brings back the 0, 1, 2... row numbers.

**5. Inspecting the DataFrame**

Once you’ve created a DataFrame, Pandas gives you tools to explore it.

**a) df.head() — View the first few rows**

python

CopyEdit

print(df.head())

This helps you preview the structure and contents.

**b) df.shape — Get the number of rows and columns**

python

CopyEdit

print(df.shape)

If it returns (4, 3) — it means 4 rows and 3 columns.

**c) df.info() — Summary of the dataset**

python

CopyEdit

print(df.info())

Gives info like:

* Column names
* Number of non-null entries
* Data types (object, int, etc.)
* Memory usage

**Summary Table**

| **Command** | **Purpose** |
| --- | --- |
| pd.DataFrame() | Creates a DataFrame from a dictionary |
| .set\_index(col) | Makes a column the new row index |
| .reset\_index() | Resets index to default numbers |
| .head() | Shows first 5 rows |
| .shape | Returns (rows, columns) |
| .info() | Prints column types and memory usage |