**TRAINING REPORT**

**ON Advanced Python**

**SUMMER INTERSHIP  
Submitted to   
Faculty of Computers and Artificial Intelligence, University** **of Sadat City**

In partial fulfillment of the requirements

for the award of the degree 

**BACHELOR OF Computer Sciences**

**(Semester - IV)**

Place: **Public service center**

Date: **23/9/2023**

Submitted by: **Mohamed Hatem**

Student’s Name: **Mohamed Hatem Dawood**

Student’s Signature:

ID: **2031050**

**CONTENTS**

CHAPTER 1: **INTRODUCTION**

1. Introduction to python

CHAPTER 2: **Python Basics**

2.1 Variables

2.2 Data Types

2.3 String

2.4 Operators

2.5 Conditions

2.6 Loops

2.7 Functions

2.8 Data Structures

CHAPTER3: **NumPy Library**

3.1 Creating arrays.

3.2 Arrays slicing.

3.3 Arrays methods

CHAPTER 4: **Pandas Library**

4.1 Series

4.2 DataFrames

4.3 Read CSV

4.4 Analyze Data

CHAPTER 5: **Python Matplotlib**

5.1 Plotting

5.2 Scatter

5.3 Bars

5.4 Histograms

5.5 Pie charts

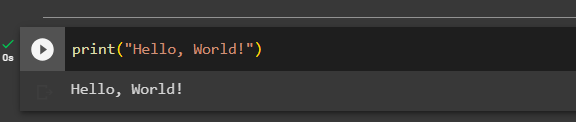
CHAPTER 6: **Librosa**

**Chapter 1: Introduction**

* Python is a popular programming language. It was created by **Guido van Rossum,** and released in 1991.

It is used for:

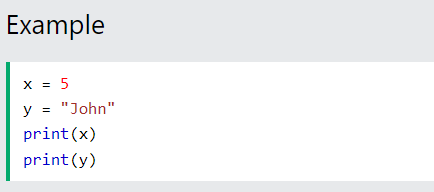
* web development (server-side),
* software development,
* mathematics,
* system scripting.
* Why Python?
* Python was designed for readability and has some similarities to the English language with influence from mathematics.
* Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
* Python relies on indentation, using whitespace, to define scope, such as the scope of loops, functions and classes. Other programming languages often use curly brackets for this purpose.
* Example:



**Chapter 2: Python Basics**

**2.1 Variables**

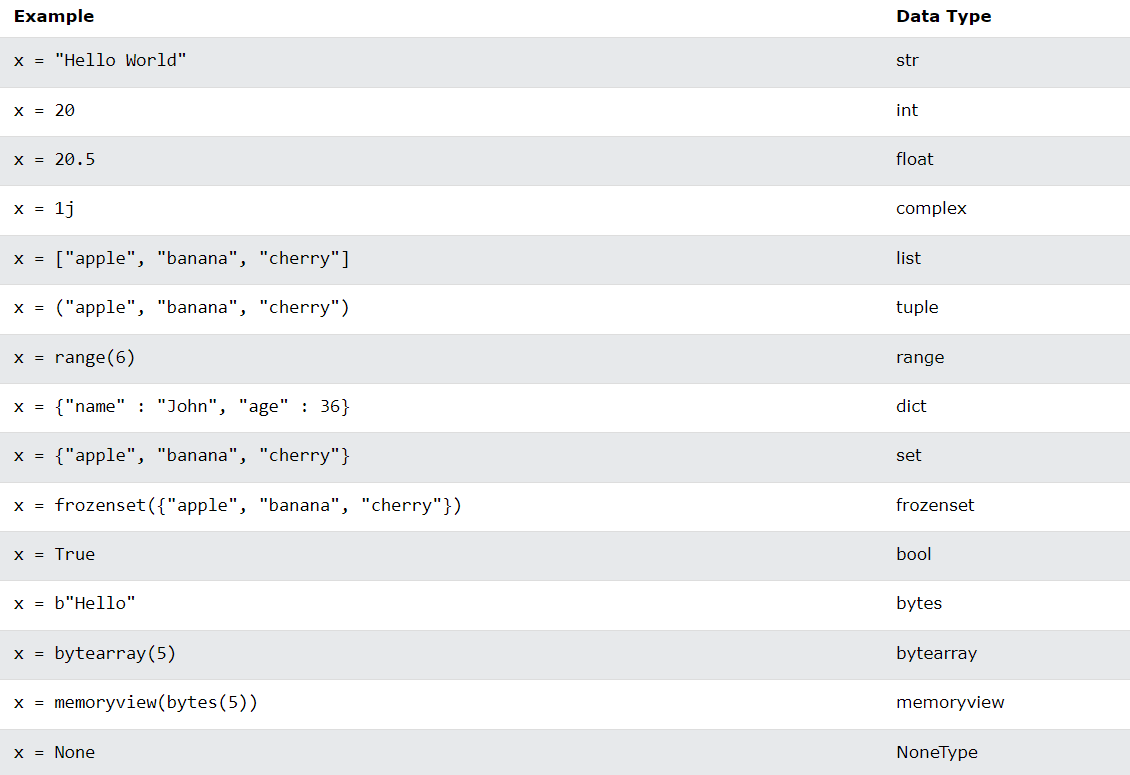
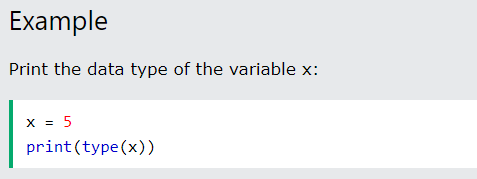
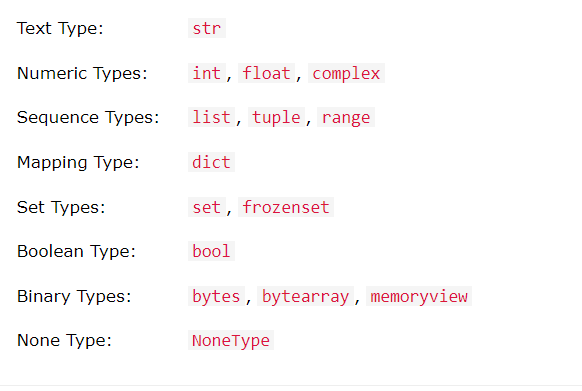
- Variables are containers for storing data values.



**2.2 Data Types**

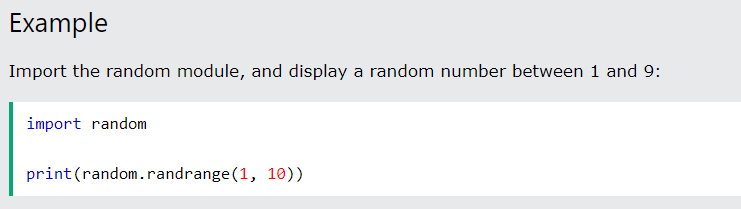
In programming, data type is an important concept.

Variables can store data of different types, and different types can do different things.



* Random number

Python does not have a random() function to make a random number, but Python has a built-in module called random that can be used to make random numbers:

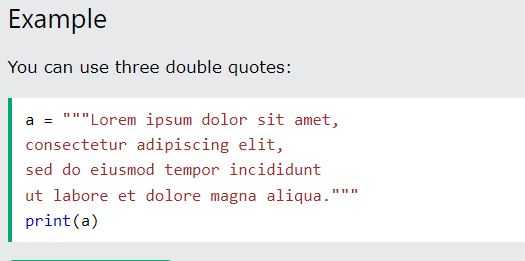


**2.3 Strings**

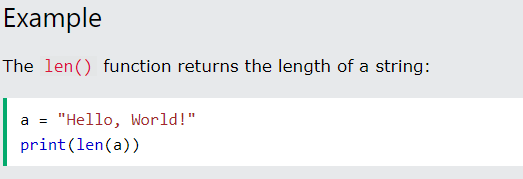
- Strings in python are surrounded by either single quotation marks, or double quotation marks.

'hello' is the same as "hello".

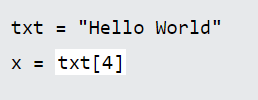
* Multiline string



* To get the length of a string, use the len() function.

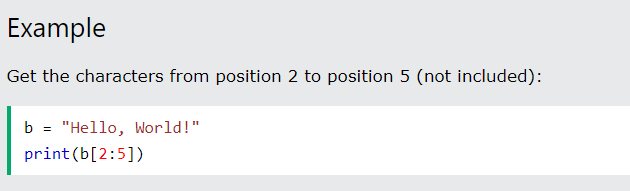


* To get the any character of string

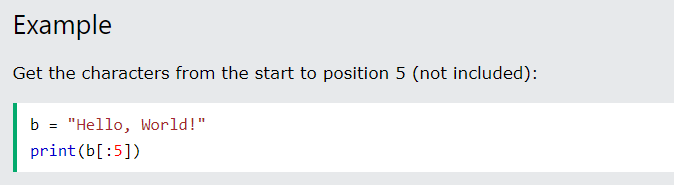


**Note:**The first character has index 0.

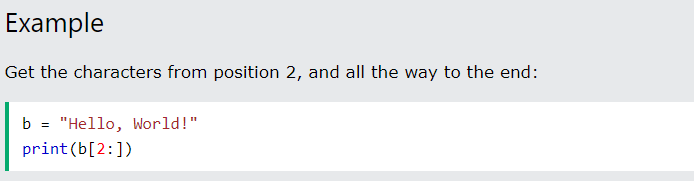
* Slicing string
  + You can return a range of characters by using the slice syntax.
  + Specify the start index and the end index, separated by a colon, to return a part of the string.



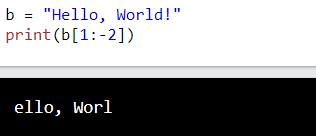
* Slice from the start.
  + By leaving out the start index, the range will start at the first character:



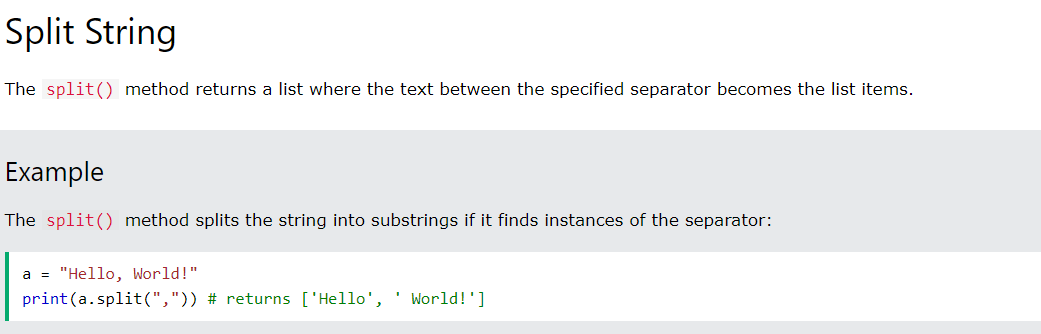
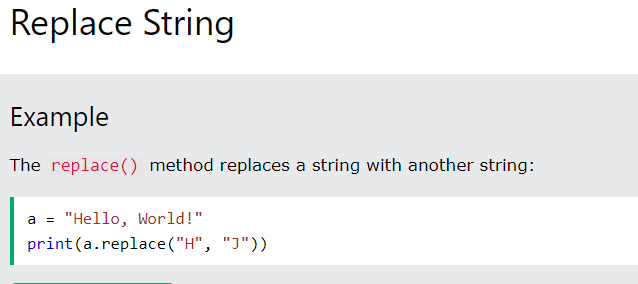
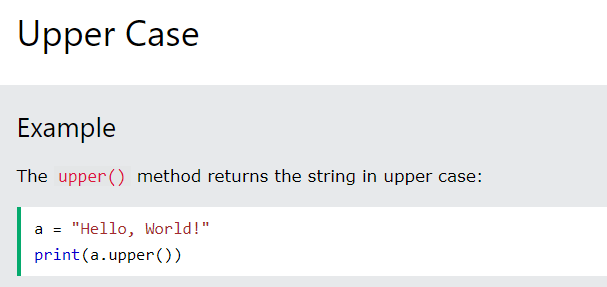
* Slice to the end.
  + By leaving out the end index, the range will go to the end:



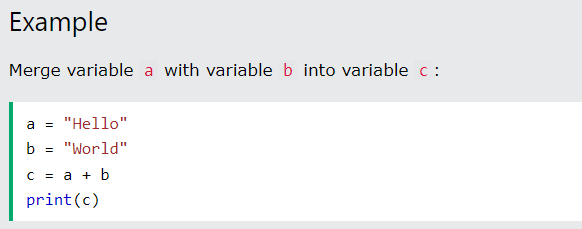
* Negative indexing.
  + Use negative indexes to start the slice from the end of the string:



* Python has a set of built-in methods that you can use on strings.
* **Note:** All string methods return new values. They do not change the original string.



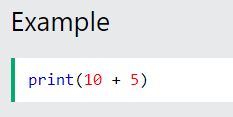
* Concatenate strings
  + To concatenate, or combine, two strings you can use the + operator.



**2.4 Operators**

- Operators are used to perform operations on variables and values.

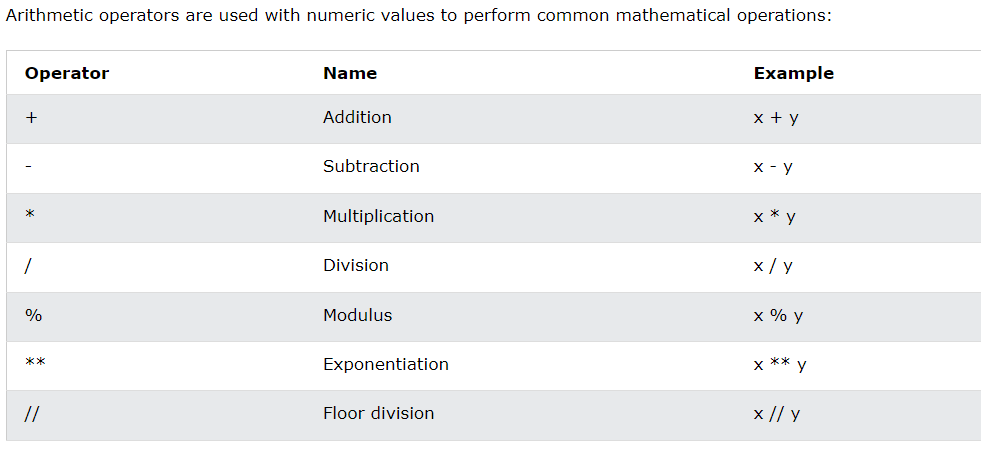
In the example below, we use the + operator to add together two values:



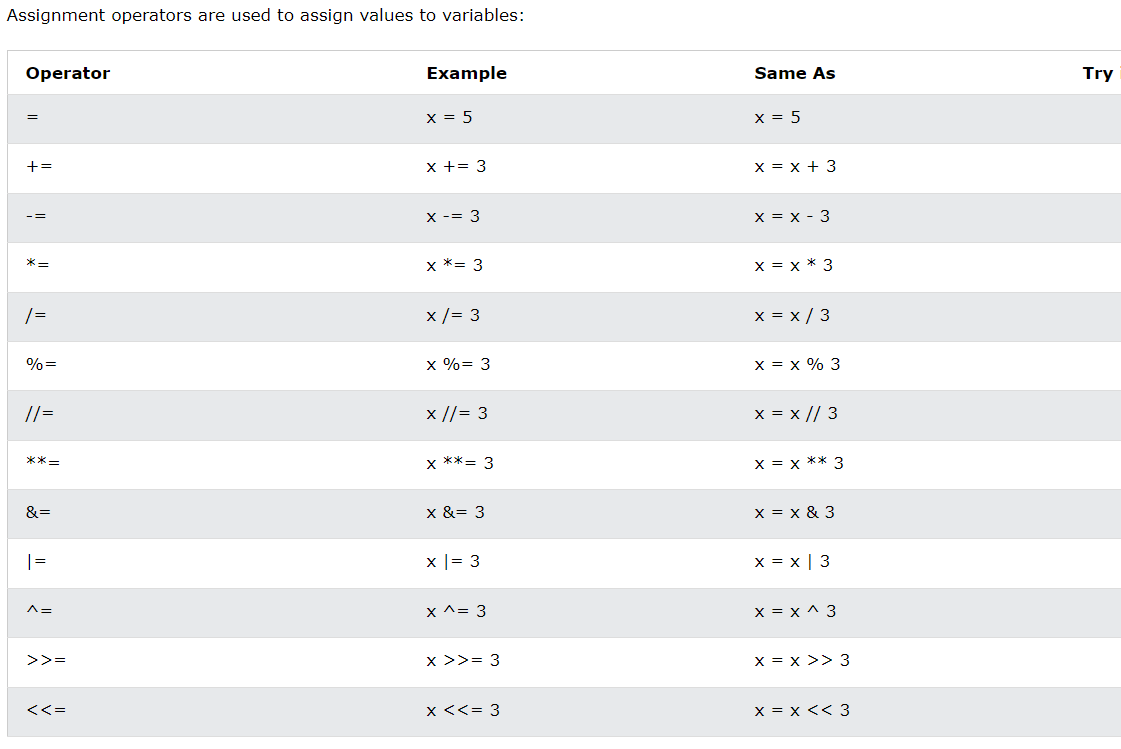
Python divides the operators in the following groups:

* Arithmetic operators
* Assignment operators
* Comparison operators
* Logical operators
* Identity operators
* Membership operators
* Bitwise operators

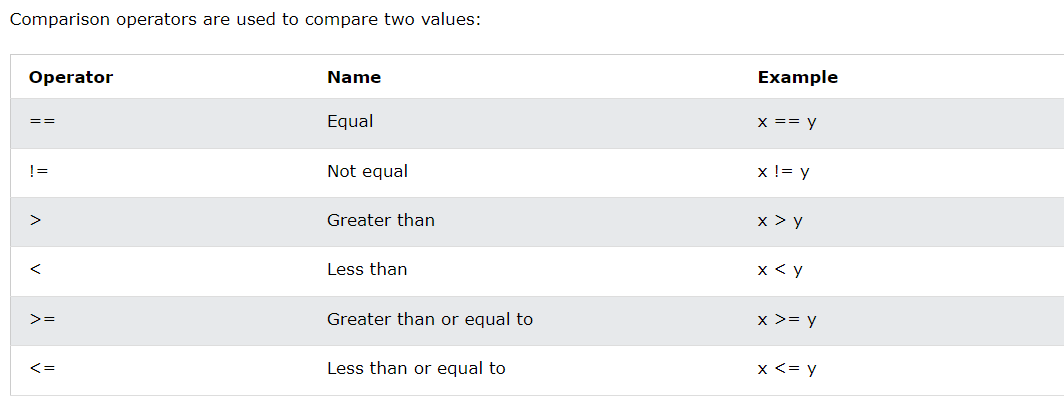
Arithmetic operators:



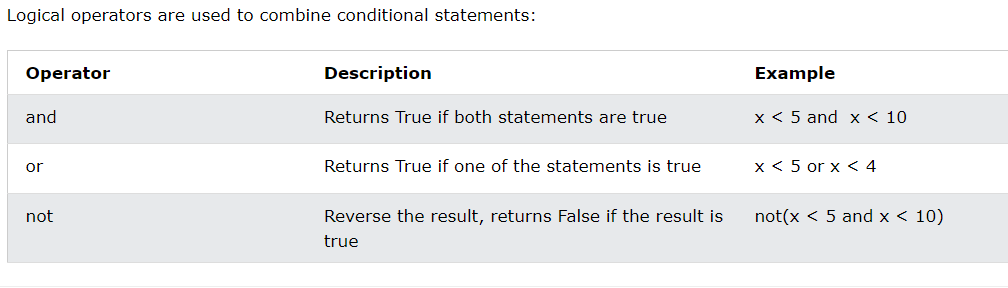
Assignment operators:



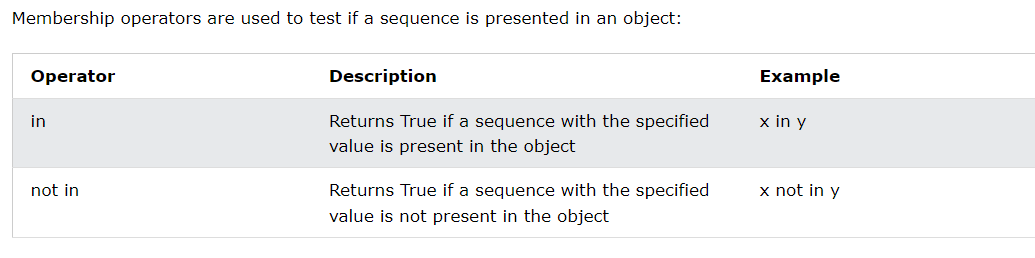
Comparison operators:



Logical operators:



Membership operators:

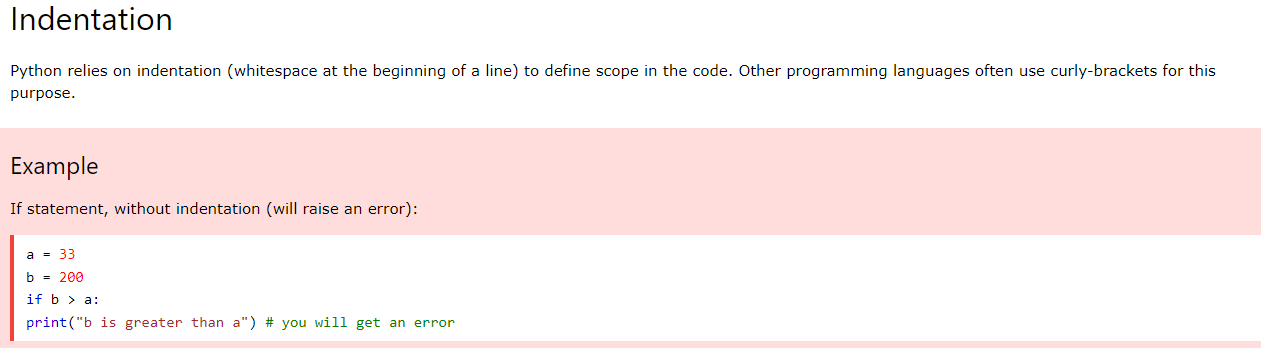
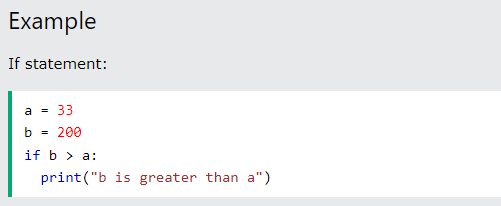


**2.5 Conditions**

Python supports the usual logical conditions from mathematics:

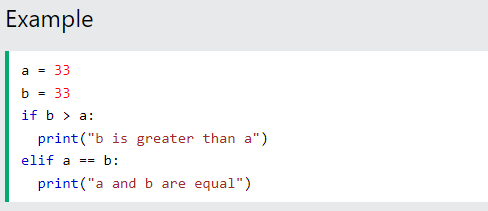
* Equals: a == b
* Not Equals: a != b
* Less than: a < b
* Less than or equal to: a <= b
* Greater than: a > b
* Greater than or equal to: a >= b

An "if statement" is written by using the if keyword.



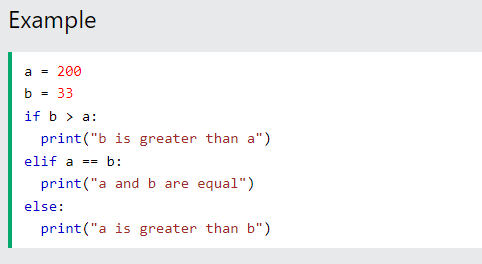
## Elif

The elif keyword is Python's way of saying "if the previous conditions were not true, then try this condition".



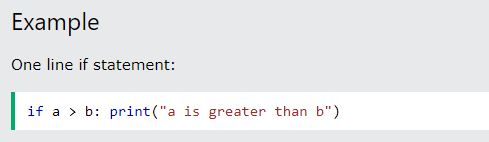
## Else

The else keyword catches anything which isn't caught by the preceding conditions.



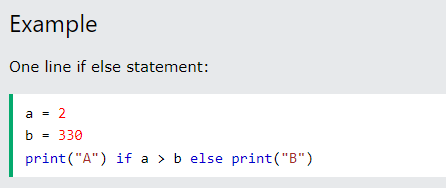
## Shorthand If

If you have only one statement to execute, you can put it on the same line as the if statement.



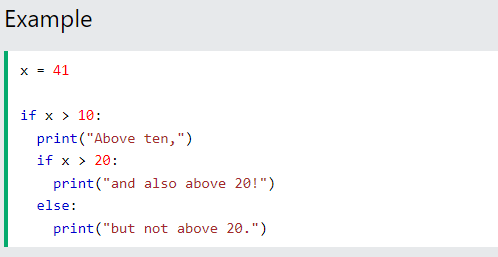
## Short Hand If ... Else

If you have only one statement to execute, one for if, and one for else, you can put it all on the same line:

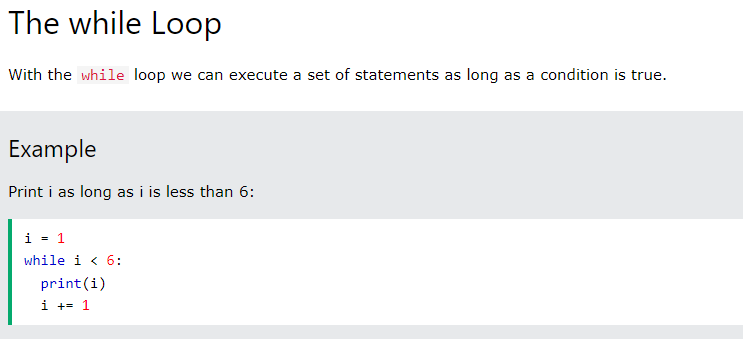
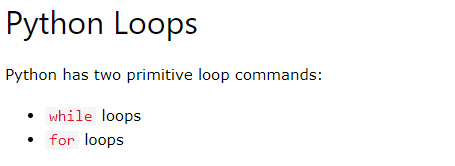


## Nested If

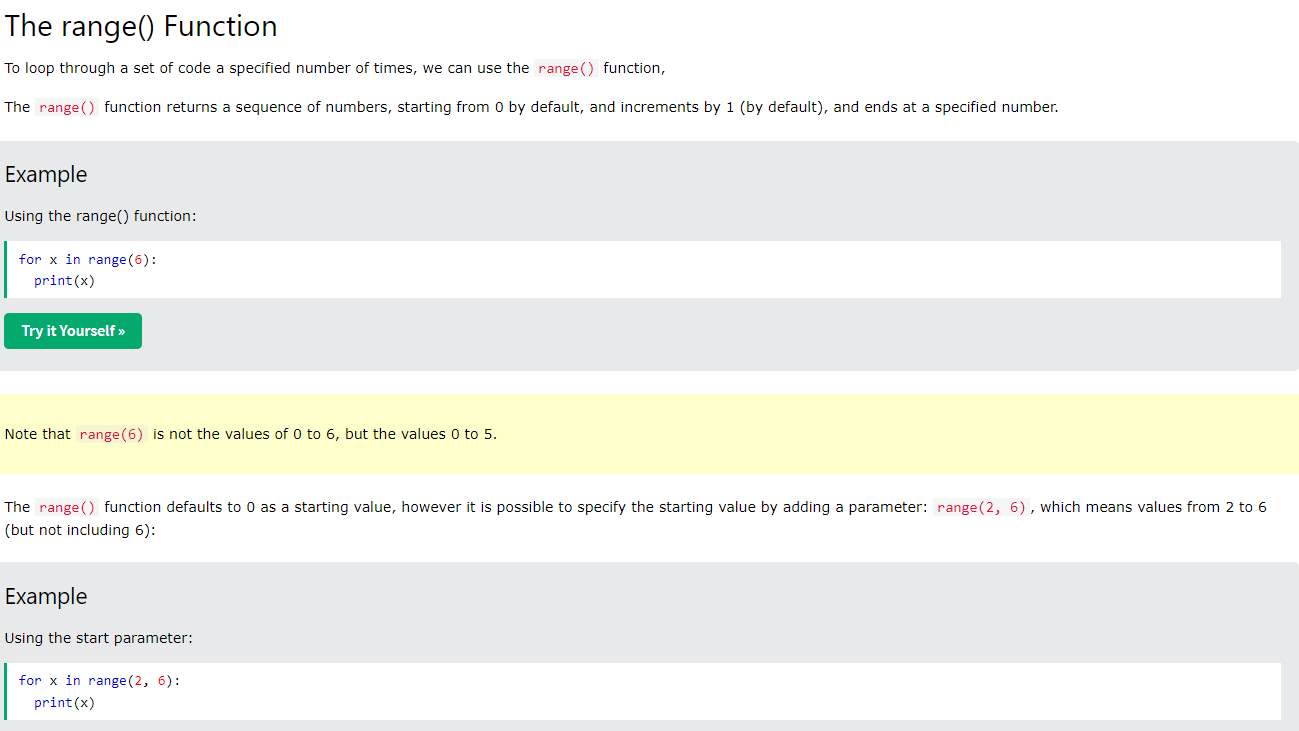
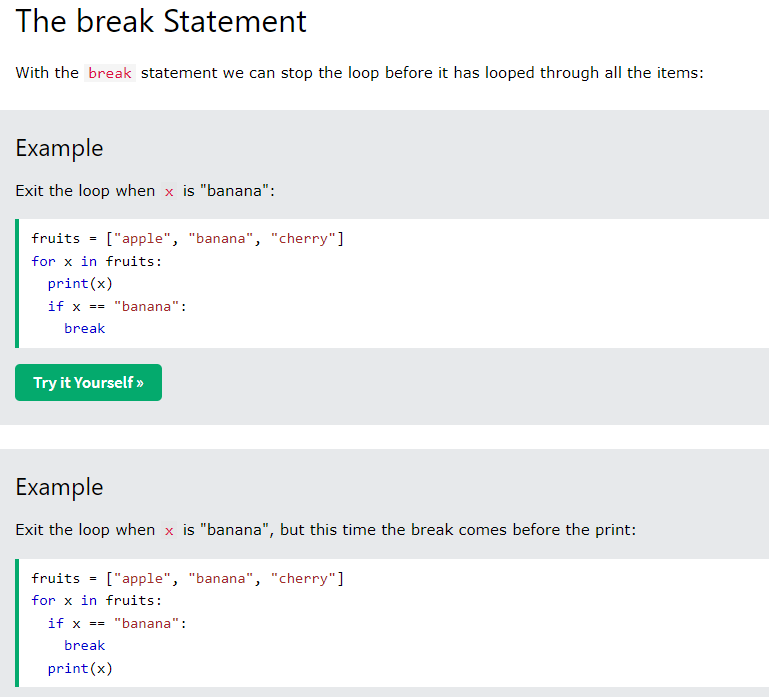
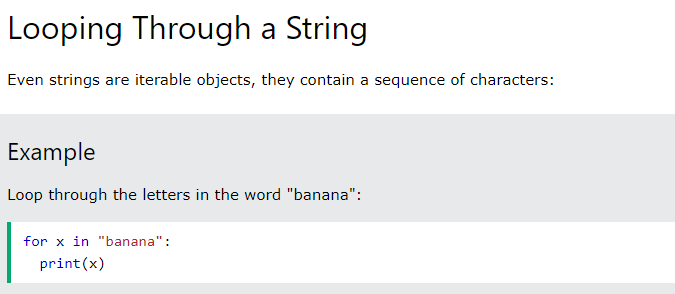
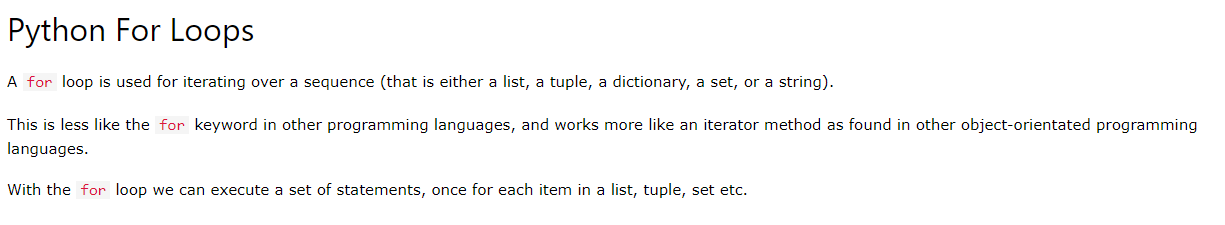
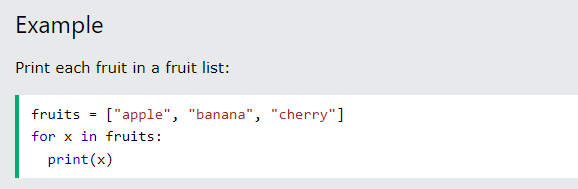
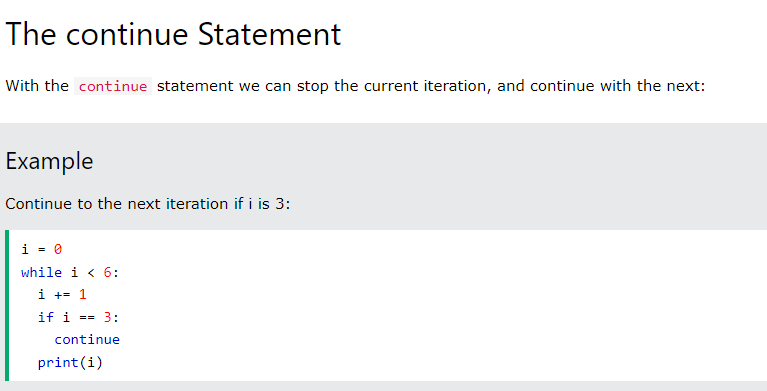
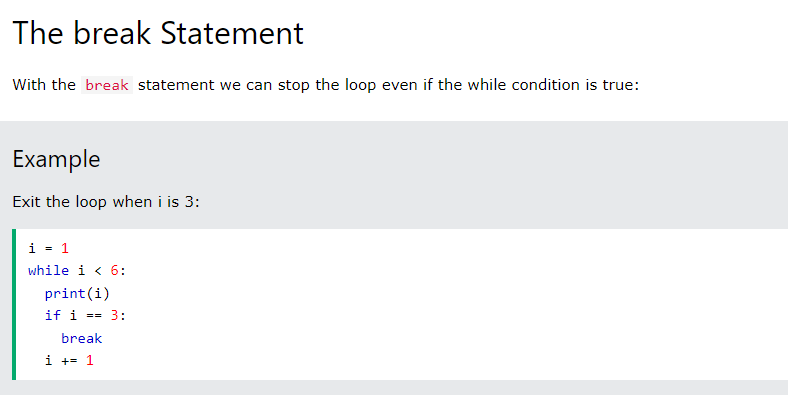
You can have if statements inside if statements, this is called nested if statements.



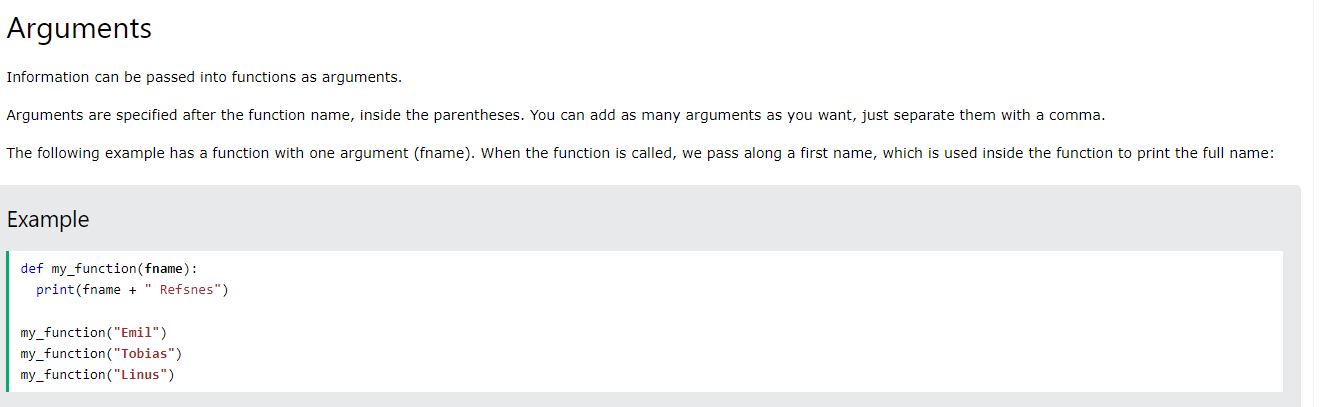
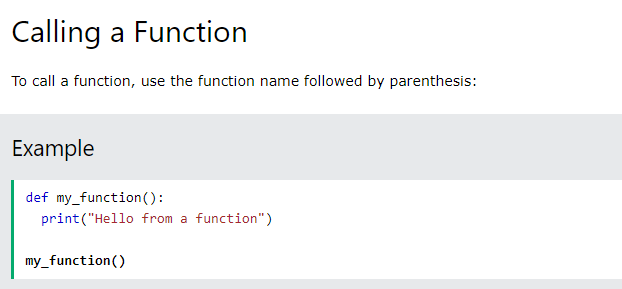
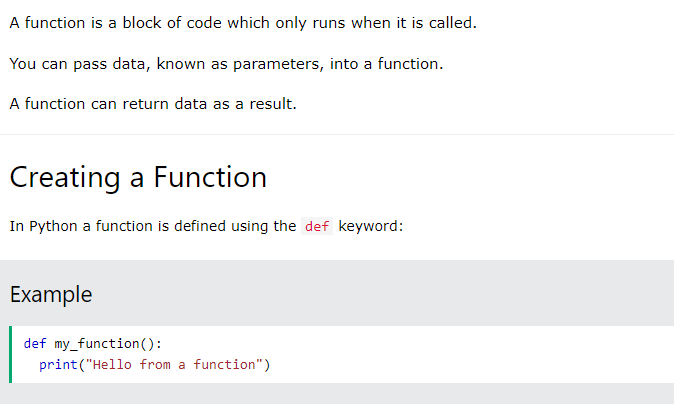
**2.7 Loops**



**Note:** remember to increment i, or else the loop will continue forever.



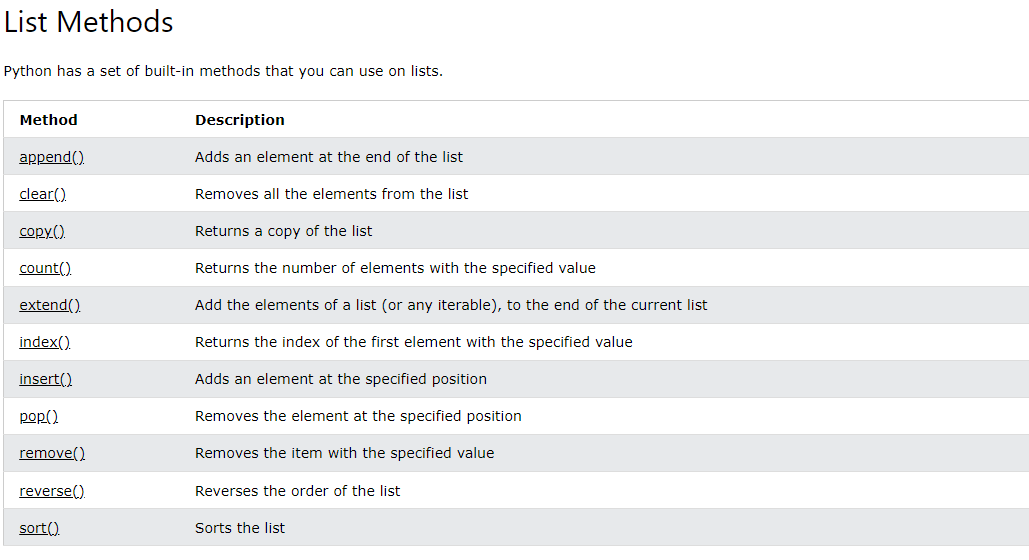
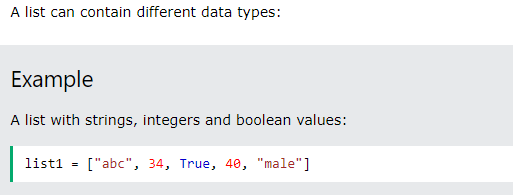
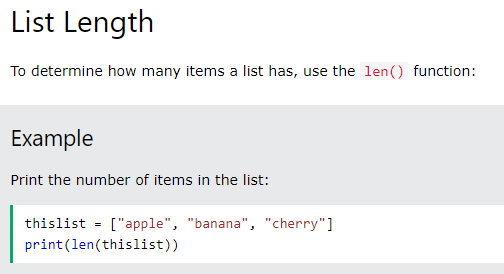
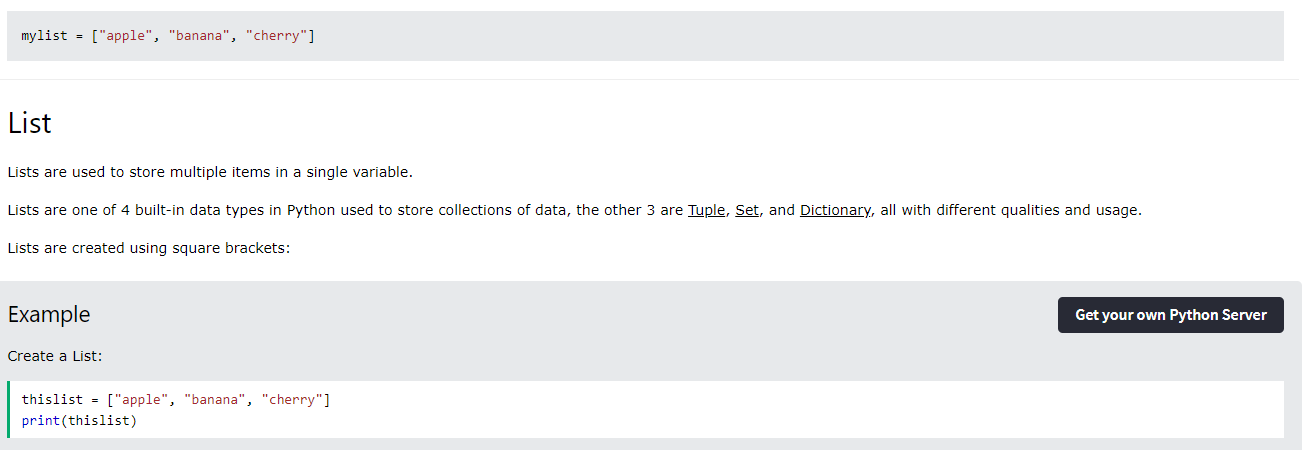
**2.7 Functions**



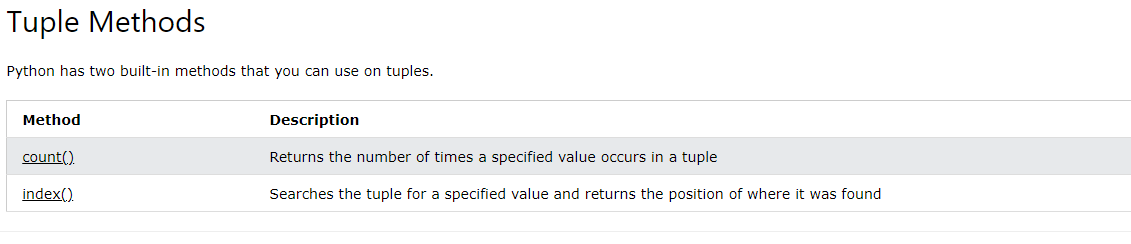
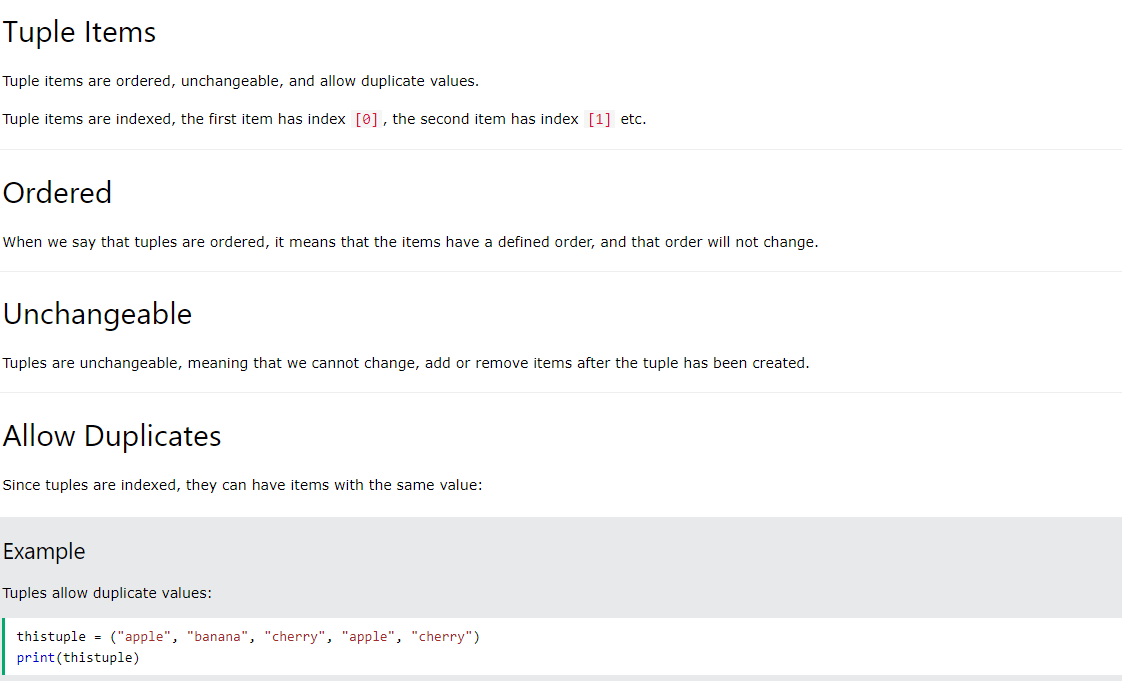
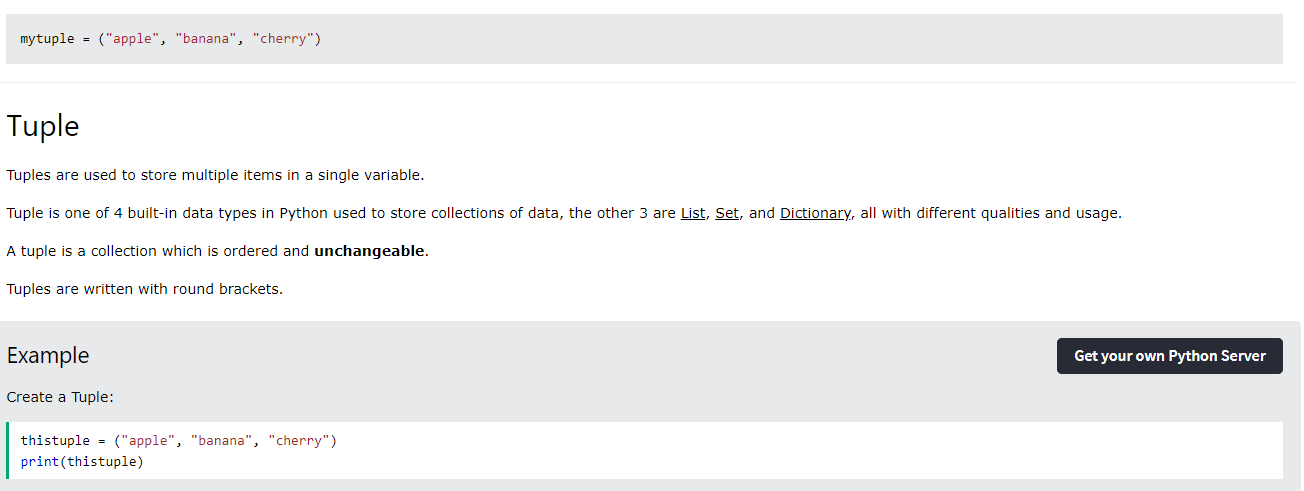
**2.8 Data structures**

**List, tuple, set, dictionary**

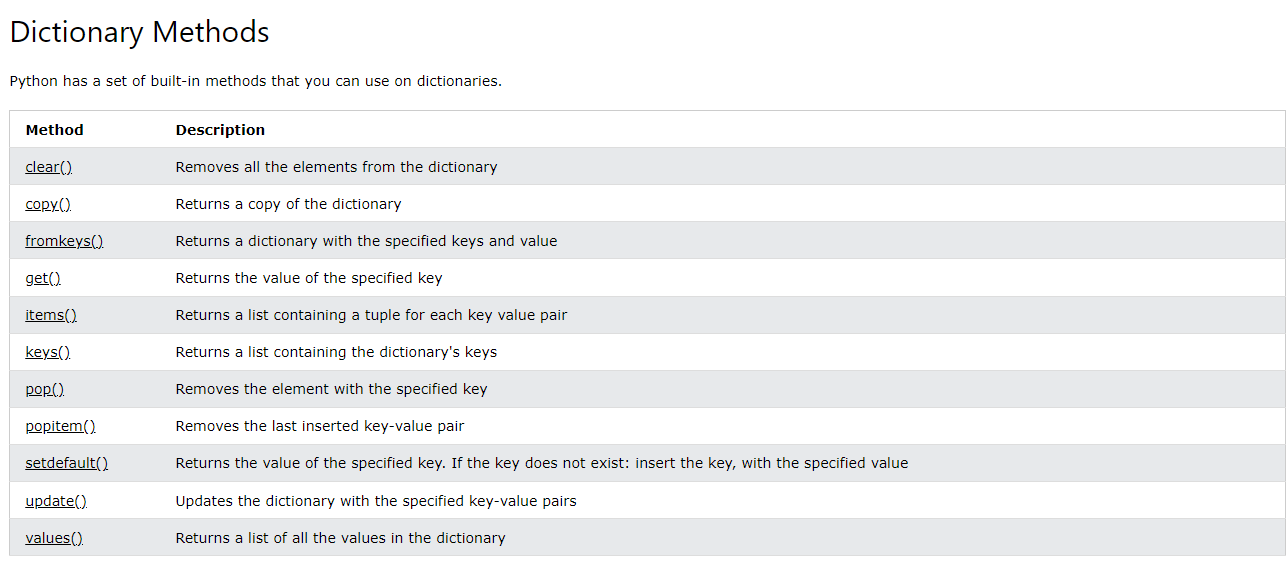
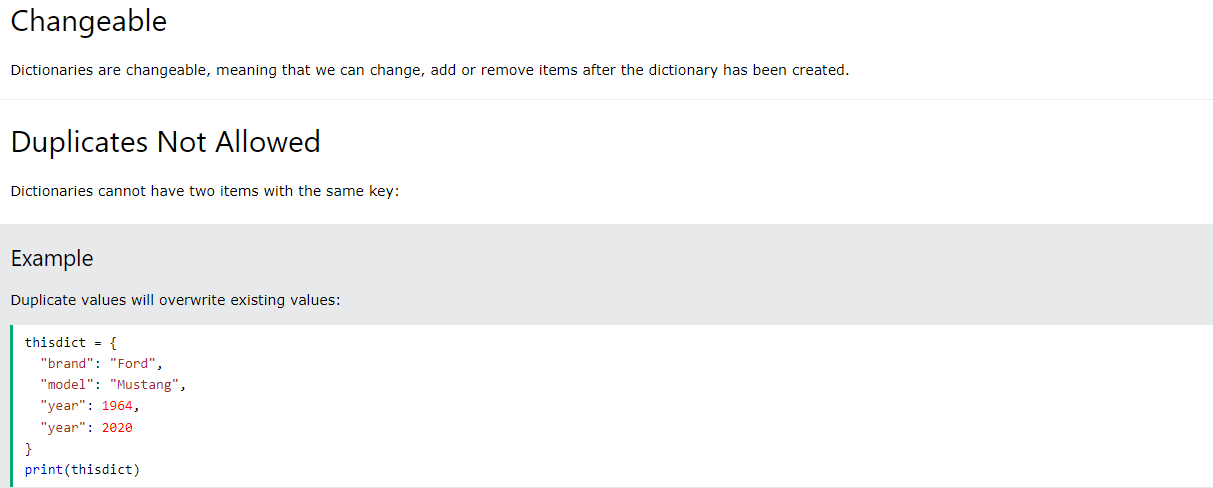
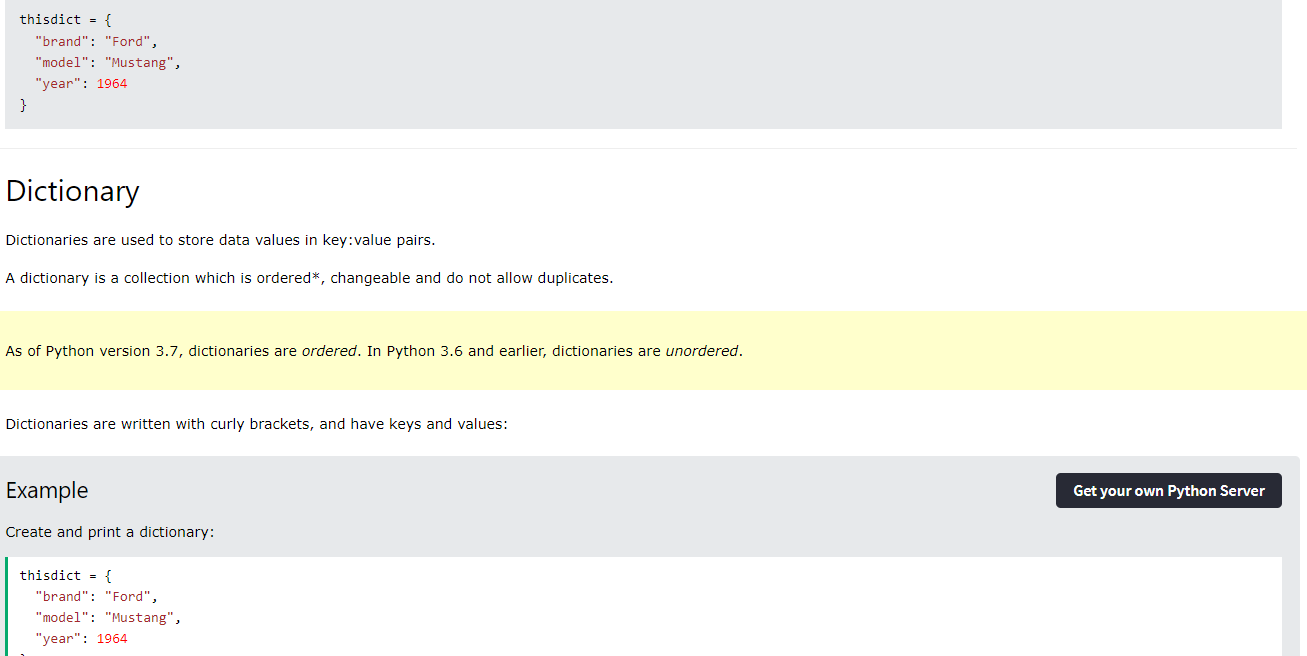
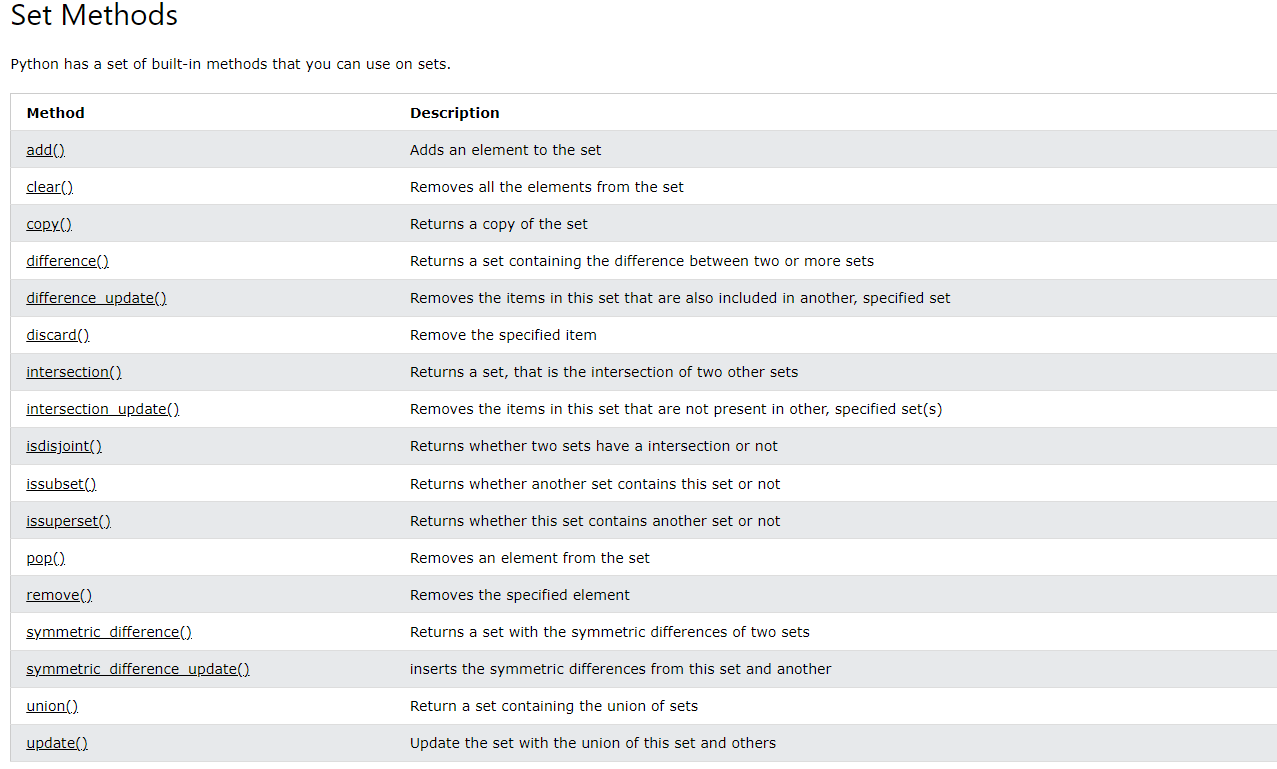
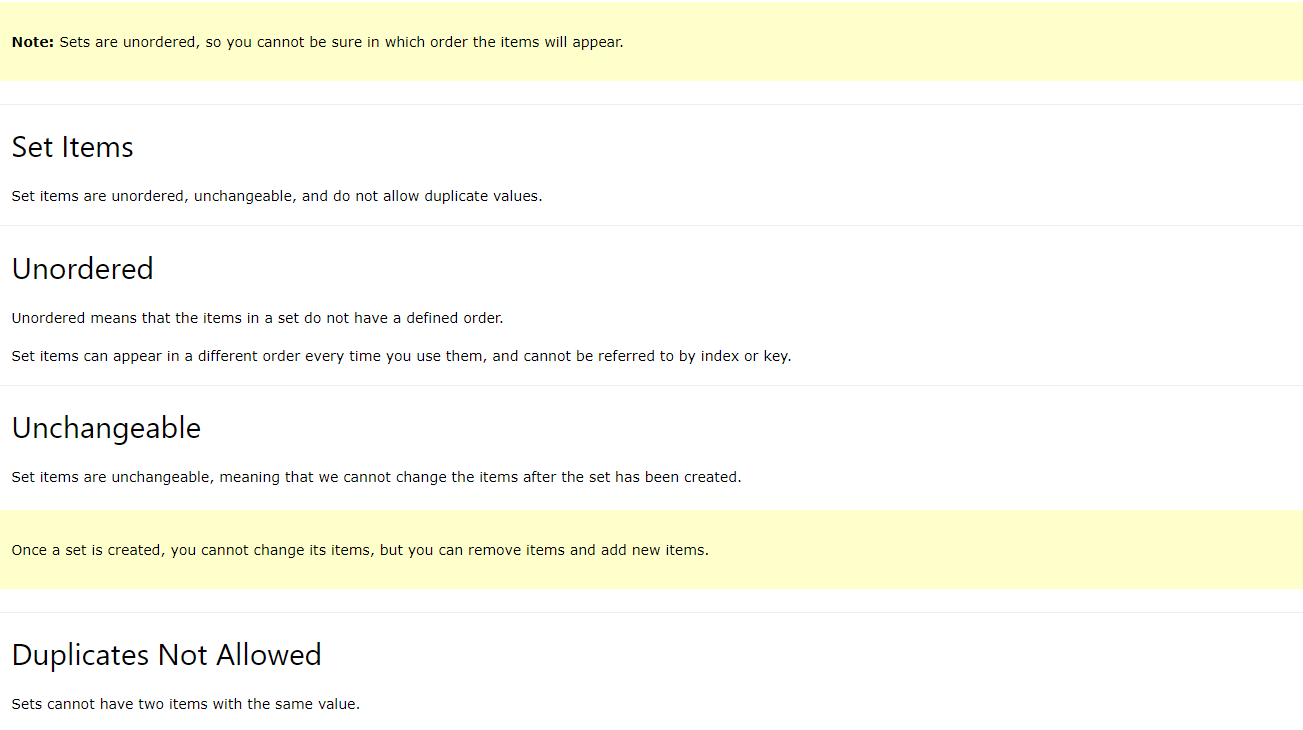
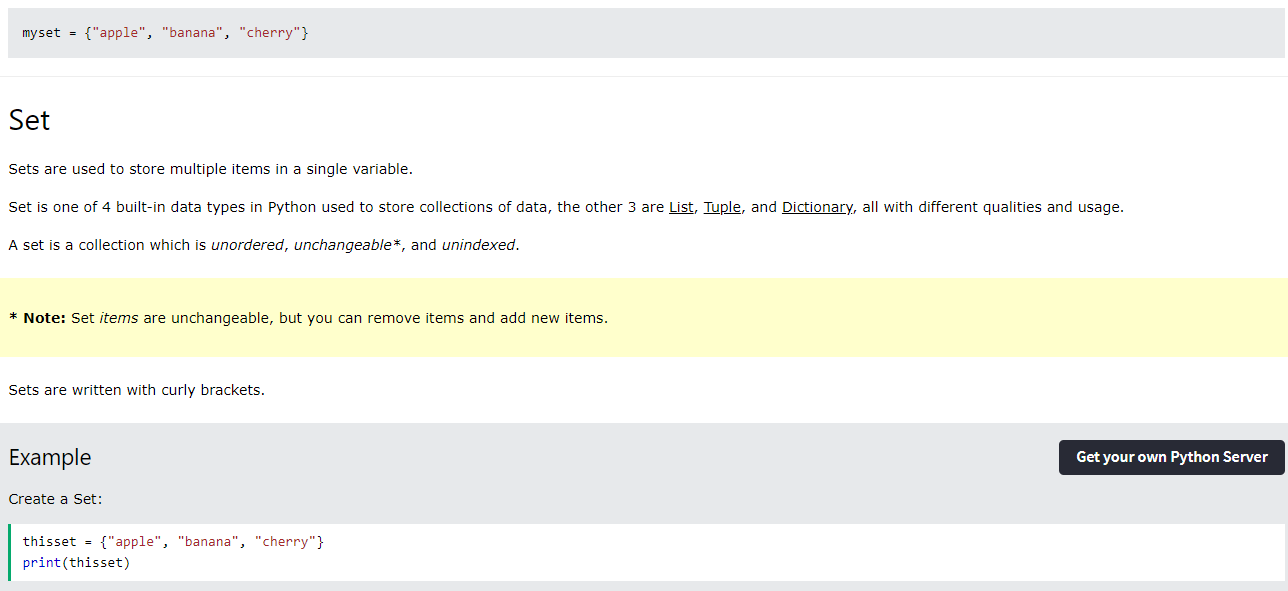
**List:**



**Tuple:**



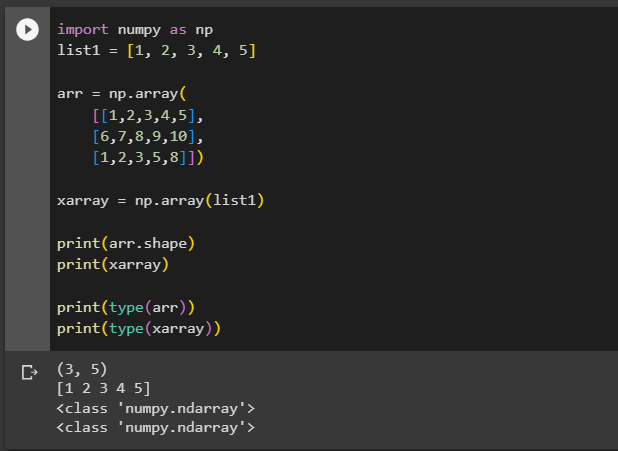
**Set:**



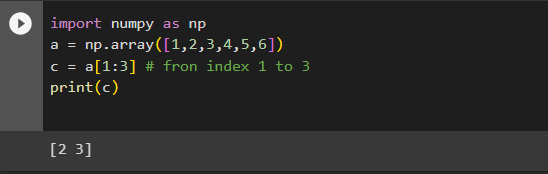
**Chapter 3: NumPy Library**

NumPy is a Python library, used for working with arrays and numeric computations.

**3.1 Creating arrays.**

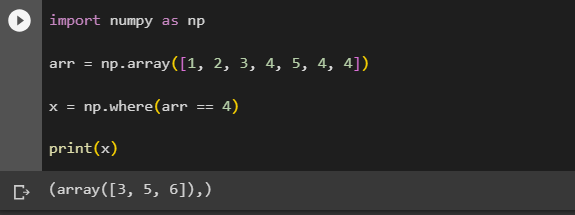


**3.2 Slicing.**

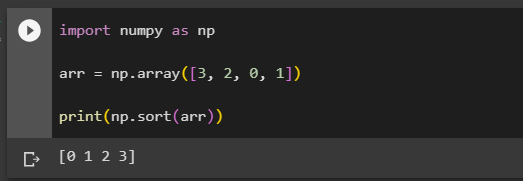


**3.3 Methods.**

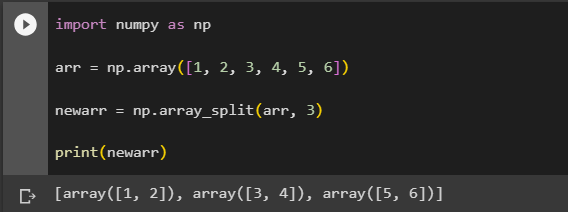
- search:



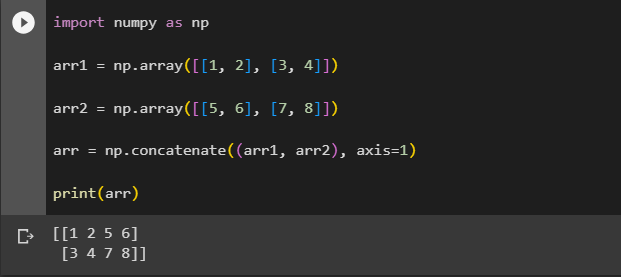
* Sort:



* Split:



* Join arrays:



**Chapter 4: Pandas Library**

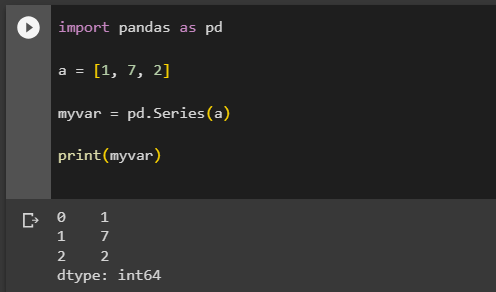
Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

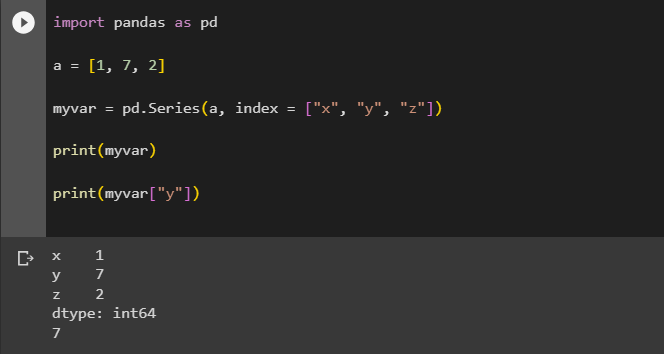
**4.1 Series**

Pandas Serie is like a column in a table.

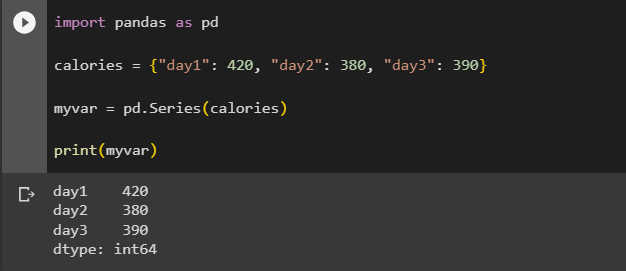
It is a one-dimensional array holding data of any type.



* To make them have labels:



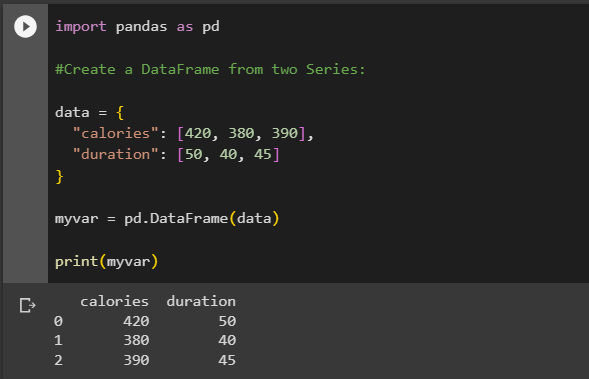
* Or directly like this:



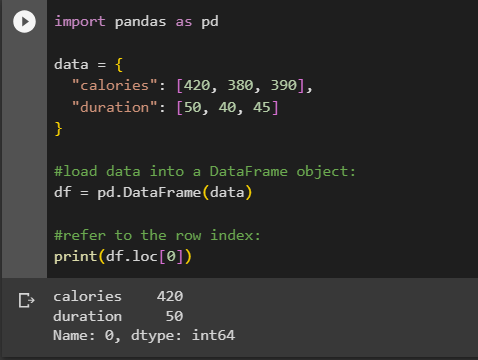
**4.2: DataFrames**

Data sets in Pandas are usually multi-dimensional tables, called DataFrames.

Series is like a column, a DataFrame is the whole table.



A Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns.

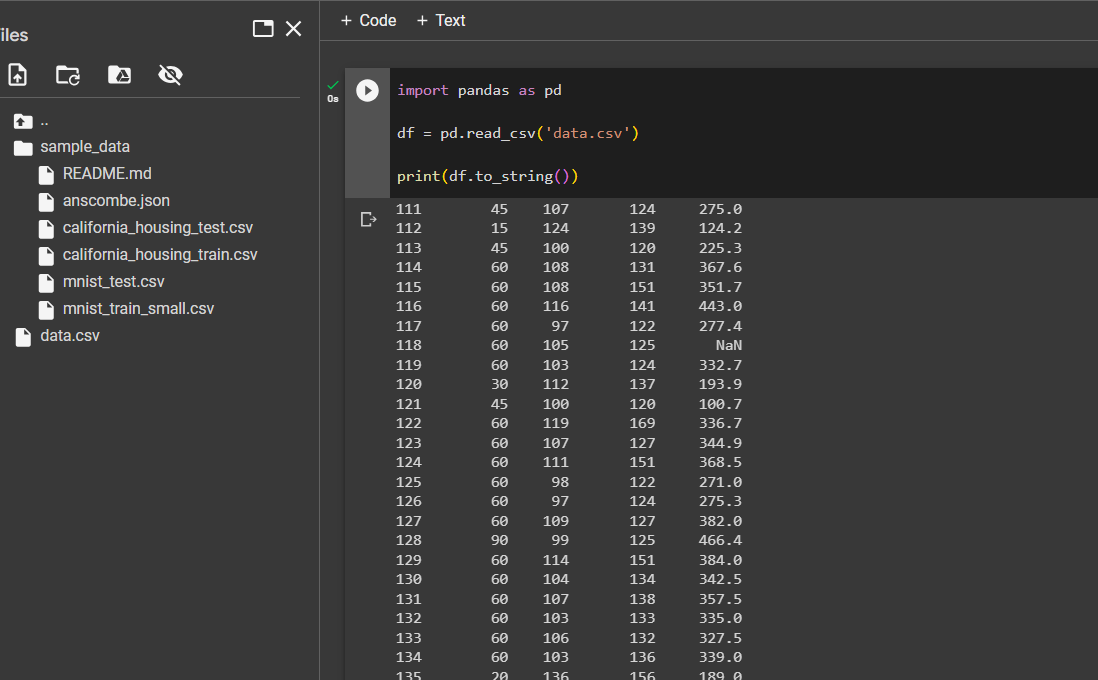


**4.3 Read CSV files**

A simple way to store big data sets is to use CSV files (comma separated files).

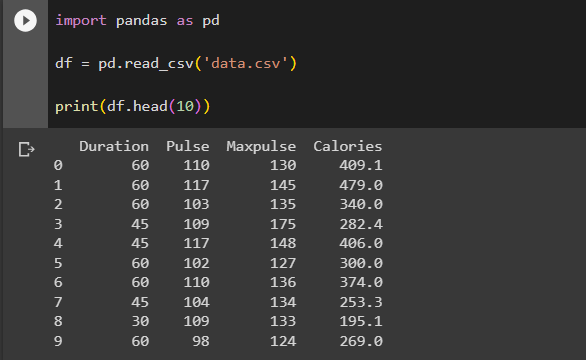
CSV files contains plain text and is a well know format that can be read by everyone including Pandas.

In our examples we will be using a CSV file called 'data.csv'.



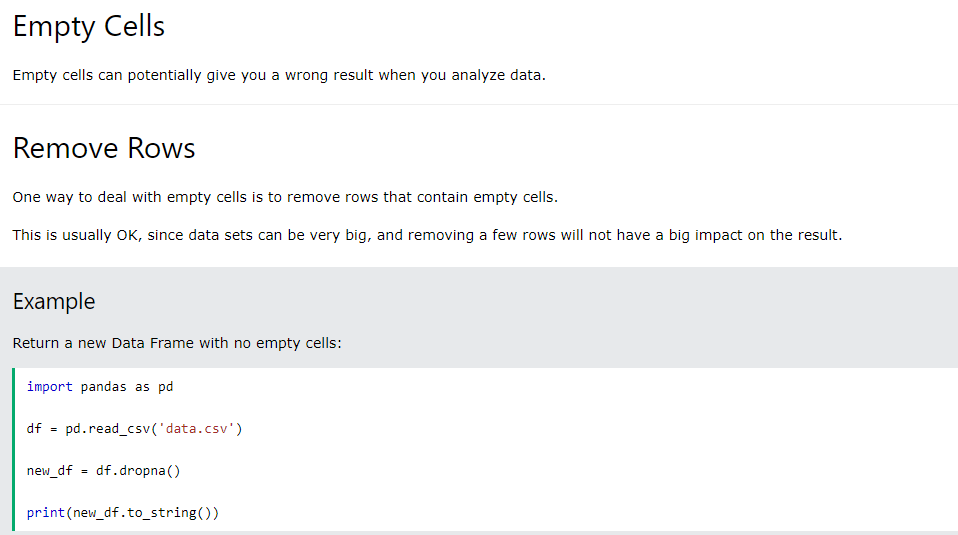
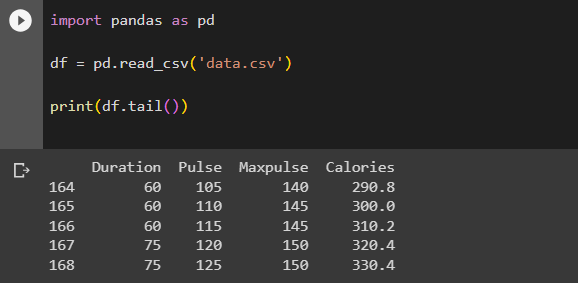
**4.4 Analyze data**

The head() method returns the headers and a specified number of rows, starting from the top.

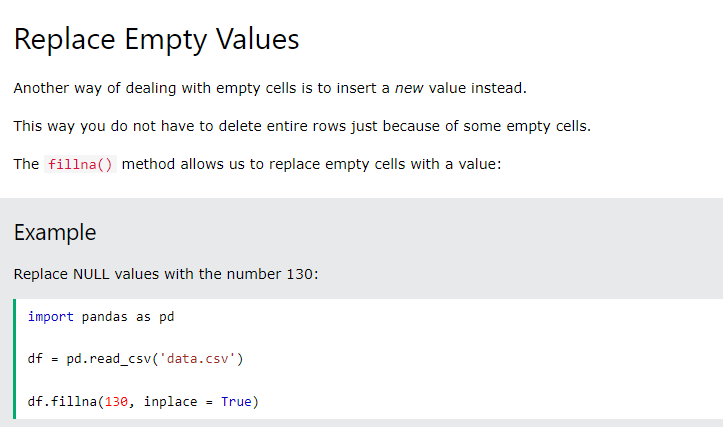
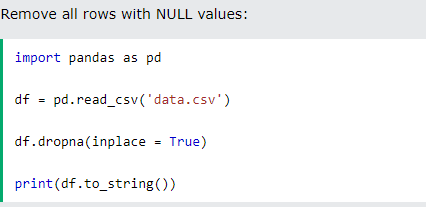


**Note:** if the number of rows is not specified, the head() method will return the top 5 rows.

The tail() method returns the headers and a specified number of rows, starting from the bottom.

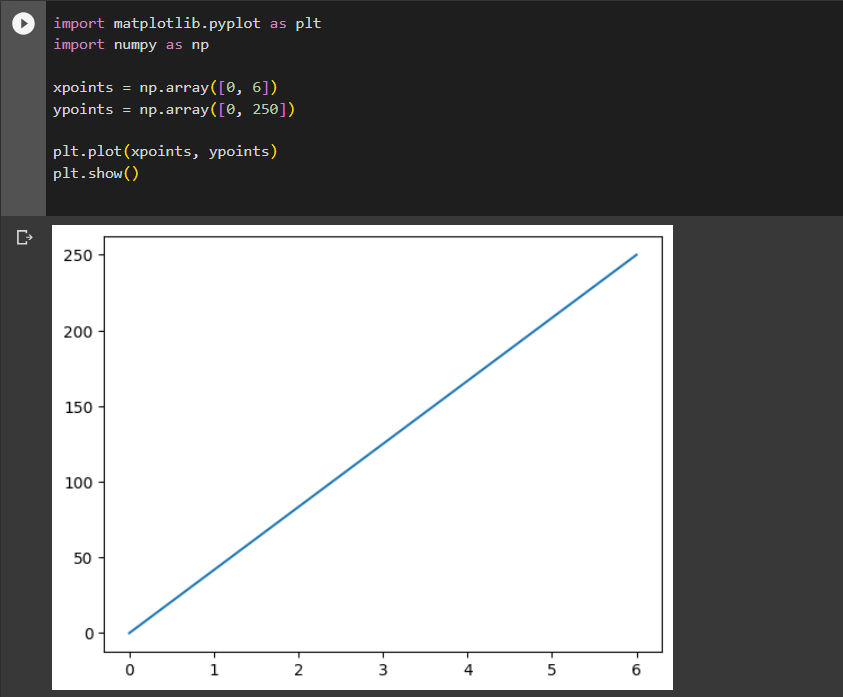


If you want to change the original DataFrame, use the inplace = True argument:



**Chapter 5: Matplotlib**

Matplotlib is a low level graph plotting library in python that serves as a visualization utility.



**5.1 Plotting**

The plot() function is used to draw points (markers) in a diagram.

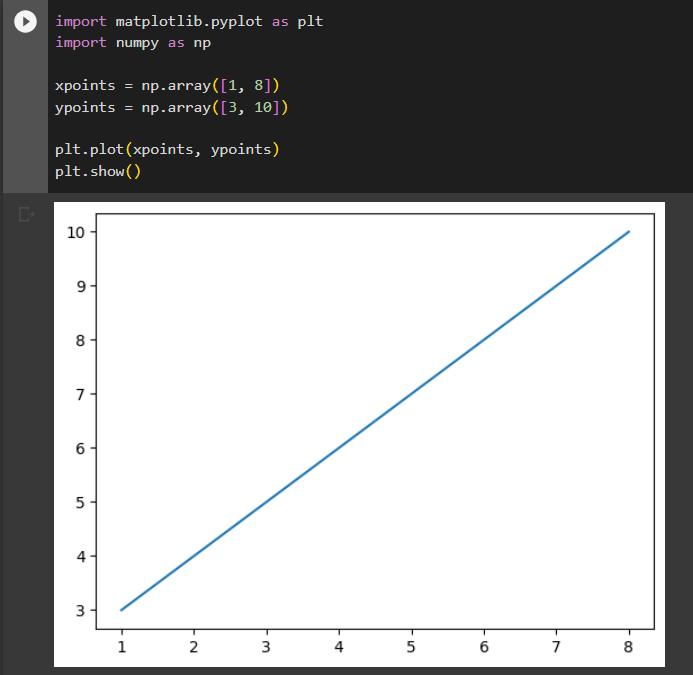
By default, the plot() function draws a line from point to point.

The function takes parameters for specifying points in the diagram.

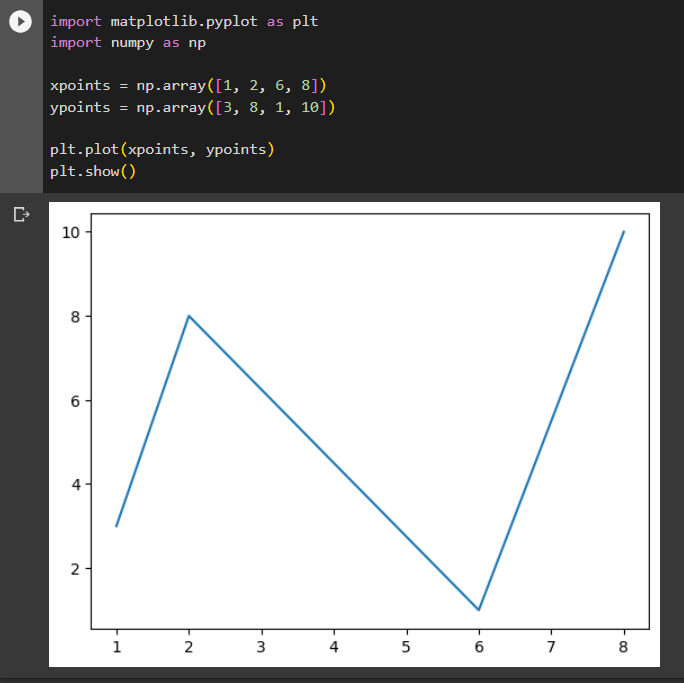
Parameter 1 is an array containing the points on the **x-axis**.

Parameter 2 is an array containing the points on the **y-axis**.

If we need to plot a line from (1, 3) to (8, 10), we have to pass two arrays [1, 8] and [3, 10] to the plot function.



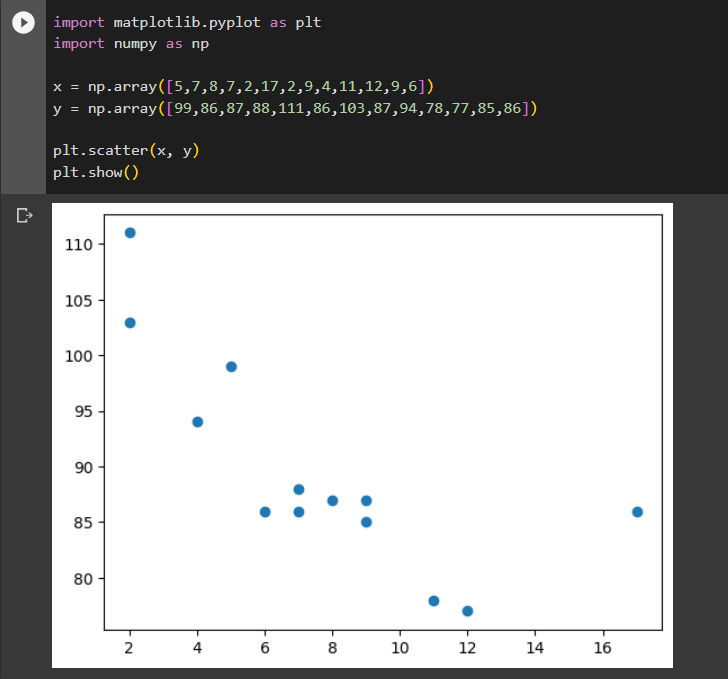
You can plot as many points as you like, just make sure you have the same number of points in both axis.



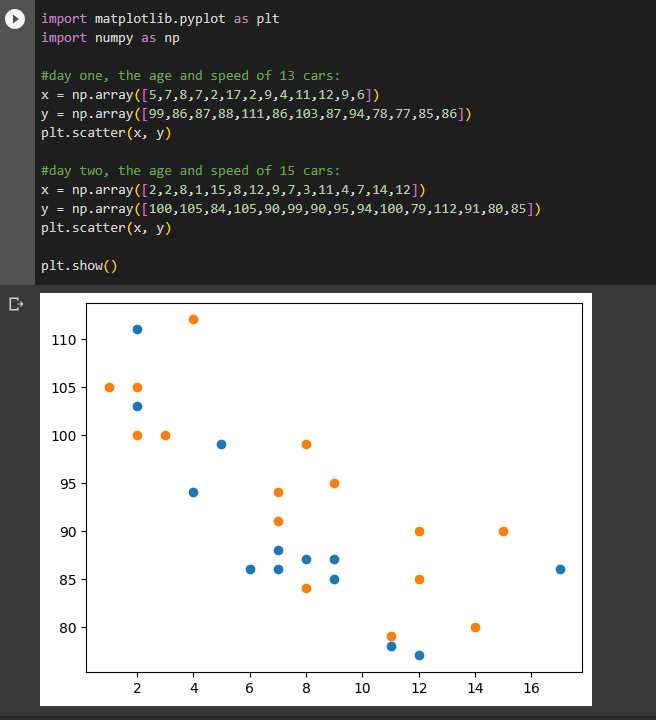
**5.2 Scatter**

With Pyplot, you can use the scatter() function to draw a scatter plot.

The scatter() function plots one dot for each observation. It needs two arrays of the same length, one for the values of the x-axis, and one for values on the y-axis:

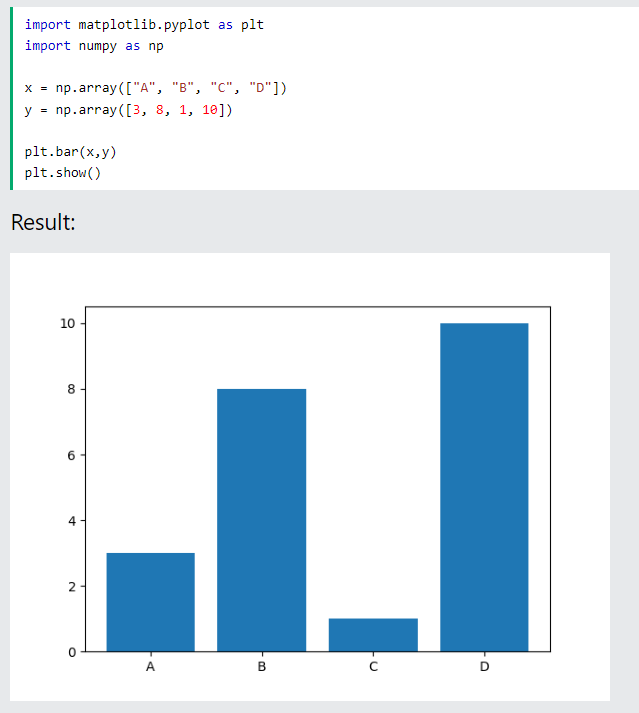


In the example above, there seems to be a relationship between speed and age, but what if we plot the observations from another day as well? Will the scatter plot tell us something else?

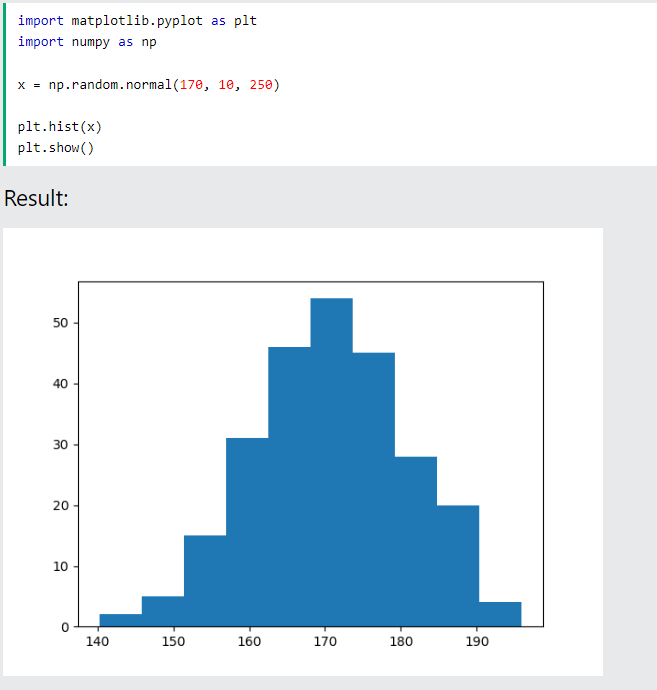


**5.3 Bars**

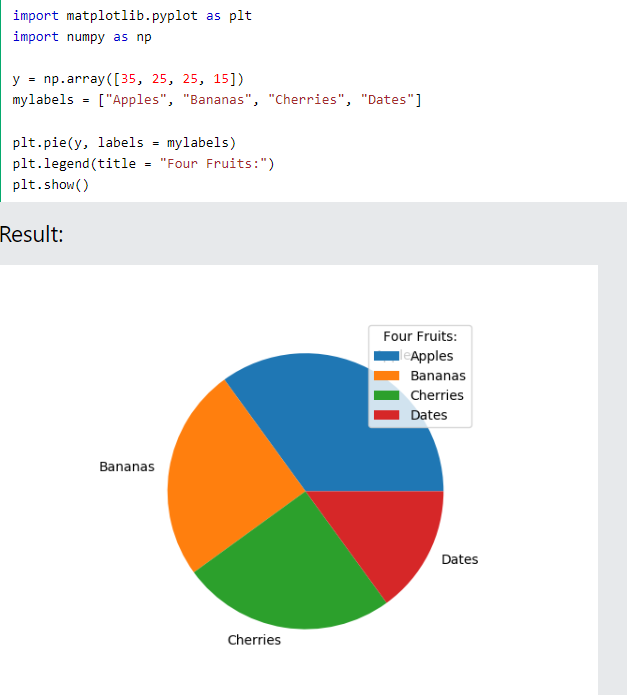
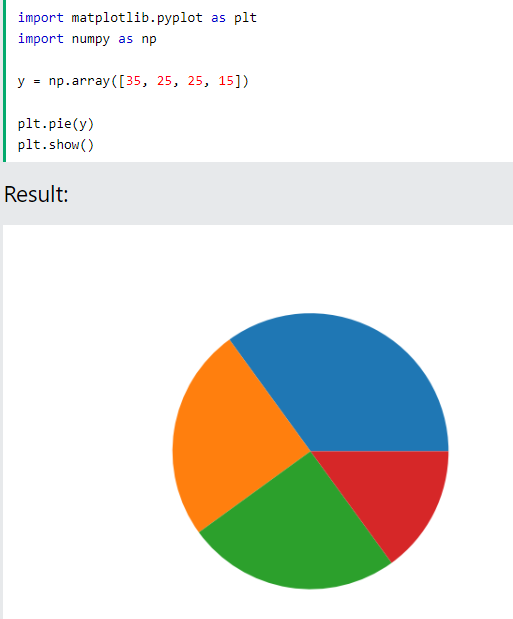
With Pyplot, you can use the bar() function to draw bar graphs:



**5.4 Histograms**



**5.5 Pie charts**



**Chapter 6: Librosa**

[**librosa**](https://librosa.org/doc/latest/core.html#module-librosa) is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.

