Python sendiri berguna dalam berbagai aspek :

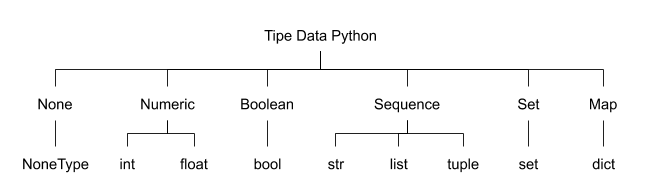
* Web Development (Server – Side)
* Software Development
* Mathematics
* Scripting
* Data Science
* Bisa mengelola Big Data dan Rumus matematika yang complex
* Cocok untuk riset dan rapid prototype suatu product dan launch hingga produksi
* CRUD sebuah file dan database

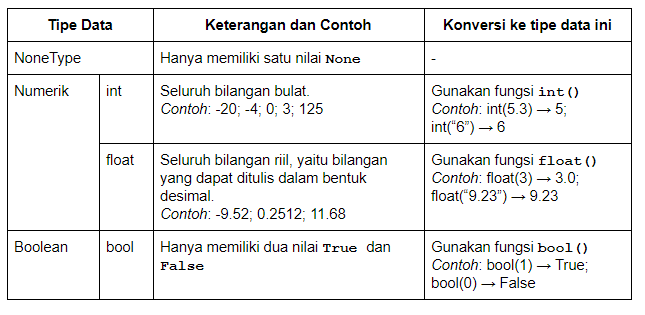
## **Struktur Bahasa Python - Part 2**

Aku menyimak tampilan layar laptop, berusaha memahami catatan Senja untuk struktur bahasa Python yang dapat aku gunakan:

|  |  |  |
| --- | --- | --- |
| **No.** | **Struktur** | **Keterangan** |
| 1 | Statements | instruksi yang diberikan secara baris per baris untuk dijalankan oleh program |
| 2 | Variables | pengindentifikasian yang  digunakan untuk menampung sebuah data atau informasi |
| 3 | Literals | data atau informasi yang digunakan untuk mengisi suatu variabel |
| 4 | Operators | simbol-simbol yang digunakan untuk mengubah nilai dari satu variabel dengan melibatkan satu atau lebih variabel dan literal. |
| 5 | Reserved Words | kumpulan kata-kata yang memiliki makna khusus dalam bahasa pemrograman Python dan tidak dapat digunakan untuk variables dan literals |
| 6 | Whitespace | pada bahasa Python, **spasi**dan **tab**memiliki makna khusus untuk menandai serangkaian blok dalam kode Python |
| 7 | Comments | merupakan sekumpulan teks yang dituliskan di dalam sebuah program yang tidak akan mempengaruhi hasil dari sebuah program |

## **Tipe Data Python**



****

Tabel berikut menjelaskan keterangan dari tipe data**sequence**, **set**, dan **map:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipe Data** | | **Keterangan dan Contoh** | **Konversi ke tipe data ini** |
| Sequence | str | Tipe data teks yang dapat berupa huruf, kata, frasa, kalimat atau paragraf yang diapit oleh ‘ atau “  Contoh:  “a”; ‘b’; ‘saya’; “Belajar Python”; “Kita makan nasi goreng” | Gunakan fungsi **str()**  Contoh:  str(5.3) → “5.3”;  str([1, ‘buku’]) → "[1, 'buku']" |
| list | Urutan bilangan dan teks yang diapit oleh kurung siku dan masing-masing elemennya dipisahkan dengan koma.  Contoh:  [-9.52, None, True, “saya”] | Gunakan fungsi **list()**  Contoh:  list(“buku”) → ['b', 'u', 'k', 'u'];  list((1, 2, 3)) → [1, 2, 3]  list({1, 3, 6, 3}) → [1, 3, 6] |
| tuple | Urutan bilangan dan teks yang diapit oleh kurung biasa dan masing-masing elemennya dipisahkan dengan koma.  Contoh:  (-9.52, None, True, “saya”) | Gunakan fungsi **tuple()**  Contoh:  tuple(“buku”) → ('b', 'u', 'k', 'u');  tuple((1, 2, 3)) → (1, 2, 3)  tuple({1, 3, 6, 3}) → (1, 3, 6) |
| Set | set | Urutan bilangan dan teks yang diapit oleh kurung biasa dan masing-masing elemennya dipisahkan dengan koma. Setiap elemennya bernilai unik.  Contoh:  {1, 4, 4, 3} → {1, 3, 4} | Gunakan fungsi **set()**  Contoh:  set(“buku”) → ['b', 'k', 'u'];  set((1, 2, 3)) → {1, 2, 3}  set([1, 3, 6, 3]) → {1, 3, 6} |

code

text = "Belajar Python di DQLab."

print(list(text))

print(tuple(text))

print(set(text))

output

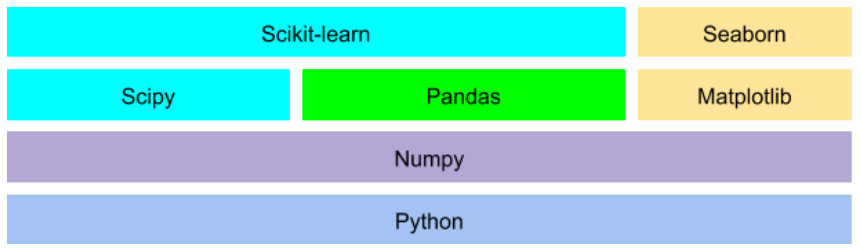
['B', 'e', 'l', 'a', 'j', 'a', 'r', ' ', 'P', 'y', 't', 'h', 'o', 'n', ' ', 'd', 'i', ' ', 'D', 'Q', 'L', 'a', 'b', '.']

('B', 'e', 'l', 'a', 'j', 'a', 'r', ' ', 'P', 'y', 't', 'h', 'o', 'n', ' ', 'd', 'i', ' ', 'D', 'Q', 'L', 'a', 'b', '.')

{'L', 'o', 'D', 'l', 'a', 'Q', 't', 'b', 'P', 'n', 'd', 'e', ' ', 'i', 'B', 'r', 'h', 'j', '.', 'y'}

list library python

* **Numpy**(numerical python) adalah library yang memudahkan dalam pendefinisian array baik 1D, 2D, 3D atau *n*D, dan juga memiliki fungsi-fungsi untuk aljabar linier.
* **Scipy**(Scientific Python) merupakan library yang ditujukan untuk keperluan komputasi saintifik seperti keperluan aljabar linier, integrasi dan diferensiasi numerik, transformasi Fourier, optimasi, interpolasi, statistik dan yang lainnya.
* **Pandas**adalah library untuk pengolahan data dalam bentuk tabular (seperti excel) yang merupakan de facto library bagi data scientist dalam mengolah data dari berbagai sumber seperti file CSV, TSV, Excel, SQL queries, Google BigQuery, SAS, Stata, SPSS, dsb.
* **Matplotlib**digunakan untuk visualisasi dari data ke dalam berbagai bentuk grafik 2D atau 3D, seperti line chart, bar chart, histogram, polar chart, error bar plot, dan jenis grafik lainnya.
* **Scikit-learn** adalah Scipy Toolkit yang ditujukan untuk menghasilkan model predictive dengan menggunakan machine learning.
* **Seaborn**merupakan library yang dibuat dari matplotlib yang ditujukan oleh visualisasi grafik statistik dengan warna yang menawan, terintegrasi dengan baik dengan pandas.



Beberapa hal yang mungkin kelebihan dari python:

* Python dirancang untuk mudah dibaca, dan memiliki beberapa kesamaan dengan bahasa Inggris dengan pengaruh dari matematika.
* Python menggunakan baris baru untuk mengakhiri perintah, dibandingkan dengan bahasa pemrograman lain yang sering menggunakan titik koma atau tanda kurung.
* Python bergantung pada indentasi, menggunakan spasi, untuk mendefinisikan ruang lingkup; seperti lingkup loop, fungsi, dan kelas. Bahasa pemrograman lainnya sering menggunakan kurung kurawal untuk tujuan ini
* Jika kalian mencari di internet tentang programming python, biasanya akan ada 2 jenis python, python 2 dan python 3. versi utama Python yang paling baru adalah Python 3, yang akan kami gunakan dalam platform ini. Meskipun begitu, Python 2. Python 2 masih banyak digunakan karena versi python2 ini dianggap sudah stable dan masih bisa digunakan untuk production level. Berbeda dengan python3 yang masih berkembang terus menerus.
* Penggunaan variabel atau suatu objek yang bisa merepresentasikan sebuah nilai atau value sangat penting dalam bahasa pemrograman. Selain itu mempermudah dalam membaca source code, pemberian variable yang efisien juga akan membuat code berjalan optimal dan dinamis. Pada sesi kali ini kita akan belajar bagaimana cara inisialisasi variable dalam beberapa data type dan menampilkannya (print).

| **Tipe Data** | **Contoh** | **Penjelasan** |
| --- | --- | --- |
| Boolean | True atau False | Menyatakan benar True yang bernilai 1, atau salah False yang bernilai 0 |
| String | "Ayo belajar Python" | Menyatakan karakter/kalimat bisa berupa huruf angka, dll (diapit tanda " atau ') |
| Integer | 25 atau 1209 | Menyatakan bilangan bulat |
| Float | 3.14 atau 0.99 | Menyatakan bilangan yang mempunyai koma |
| List | ['xyz', 786, 2.23] | Data untaian yang menyimpan berbagai tipe data dan isinya bisa diubah-ubah |
| Tuple | ('xyz', 768, 2.23) | Data untaian yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah |
| Dictionary | {'nama': 'adi','id':2} | Data untaian yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai |

In [1]: var\_string="Belajar Python DQLAB"

var\_int=10

var\_float=3.14

var\_list=[1,2,3,4]

var\_dict={"nama":"Ali",'umur':20}

var\_tuple=("satu","dua","tiga")

print(var\_string)

print(var\_int)

print(var\_float)

print(var\_list)

print(var\_tuple)

print(var\_dict)

print(type(var\_string))

print(type(var\_int))

print(type(var\_float))

print(type(var\_list))

print(type(var\_tuple))

print(type(var\_dict))

Belajar Python DQLAB

10

3.14

[1, 2, 3, 4]

('satu', 'dua', 'tiga')

{'umur': 20, 'nama': 'Ali'}

<class 'str'>

<class 'int'>

<class 'float'>

<class 'list'>

<class 'tuple'>

<class 'dict'>

nested if pada python

i=2

if(i<7):

print("nilai i kurang dari 7")

if(i<3):

print("nilai i kurang dari 7 dan kurang dari 3")

else:

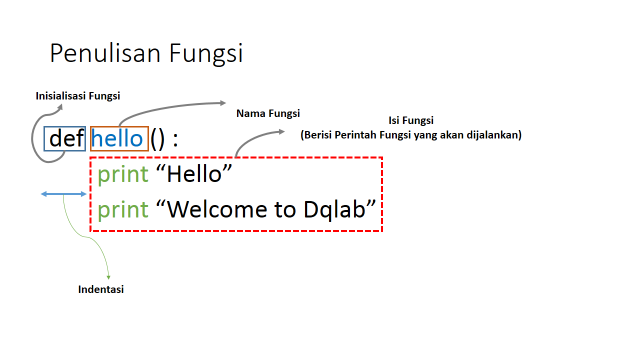
print("nilai i kurang dari 7 tapi lebih dari 3")

Keunikan lain dari looping dengan python adalah selain bahasa yang mudah dimengerti dalam looping, kita juga bisa mengakses elemen yang terdapat pada sebuah list. Berikut ini contohnya :

count=[1,2,3,4,5] #elemen list

for number in count: #looping untuk menampilkan semua elemen pada count

print("Ini adalah element count : ", number) #menampilkan elemen list pada count



contoh

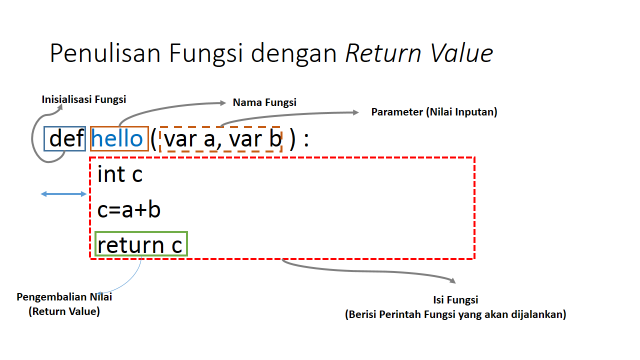
def luas\_segitiga(alas, tinggi):

luas=(alas\*tinggi)/2

print("Luas segitiga: %f" %luas)

luas\_segitiga(4,6)

%f% tipe data float



def luas\_segitiga(alas,tinggi):

luas = (alas\*tinggi)/2

return luas

print("Luas segitiga: %d"% luas\_segitiga(4,6)) %d% tipe data integer

The %d and %s string formatting "commands" are used to format strings. The %d is for numbers, and %s is for strings.

For an example:

print("%s" % "hi")

and

print("%d" % 34.6)

To pass multiple arguments:

print("%s %s %s%d" % ("hi", "there", "user", 123456)) will return hi there user123456

import math as m #menggunakan m sebagai module rename atau alias

print("Nilai pi adalah:", m.pi) #m.pi merupakan sintak untuk memanggil fungsi​

## Import Sebagian Fungsi

Pada suatu module tidak bisa dipungkiri terdiri dari puluhan bahkan ribuan fungsi. Namun, yang kita butuhkan hanya 1 atau 2 fungsi saja. Untuk meminimalisir ketidakefisienan suatu program dalam load suatu module bisa dilakukan import module namun hanya beberapa fungsi saja yang kita import kedalam code. Format **(from module\_name import function\_name)**

from math import pi

print("Nilai pi adalah", pi)

## Import Semua isi Moduls

Namun, jika memang yang dibutuhkan banyak, semisal lebih dari 10 atau bahkan ratusan fungsi, bisa dilakukan import semuanya dengan menggunakan format **from module\_name import \*.** Tanda \* disini menunjukan semua fungsi diimport kedalam code.

from math import \*

print("Nilai e adalah:", e)

import csv

#tentukan lokasi file, nama file, dan inisialisasi csv

f=open('penduduk\_gender\_head.csv','r')

reader=csv.reader(f)

#membaca baris per baris

for row in reader:

print(row)

#menutup file csv

f.close()

Bagi yang belum familiar, PANDAS merupakan salah satu library yang sangat sering digunakan untuk aplikasi dan implementasi data science. Baik untuk data manipulation, data pre-processing, atau data wrangling. Pada sesi kali ini, kita akan menggunakan PANDAS untuk membaca file dari csv.

The **head()** function is used to get the first n rows. It is useful for quickly testing if your object has the right type of data in it. For negative values of n, the **head()** function returns all rows except the last n rows, equivalent to df[:-n].

import pandas as pd

table = pd.read\_csv("https://storage.googleapis.com/dqlab-dataset/penduduk\_gender\_head.csv")

table.head()

print(table)

## Bar Chart

 Pada Python sangat banyak library untuk visualisasi. Salah satu yang paling sering digunakan adalah matplotlib karena memang sudah lama dan relatif stabil dalam perkembangannya. Maka dari itu, matplotlib dipilih untuk belajar visualisasi pada sesi kali ini. Matplotlib sendiri menyediakan banyak jenis grafik mulai dari scatter/plot, line, bar, dan lain-lain.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

table = pd.read\_csv("https://storage.googleapis.com/dqlab-dataset/penduduk\_gender\_head.csv")

table.head()

x\_label = table['NAMA KELURAHAN']

plt.bar(x=np.arange(len(x\_label)),height=table['LAKI-LAKI WNI'])

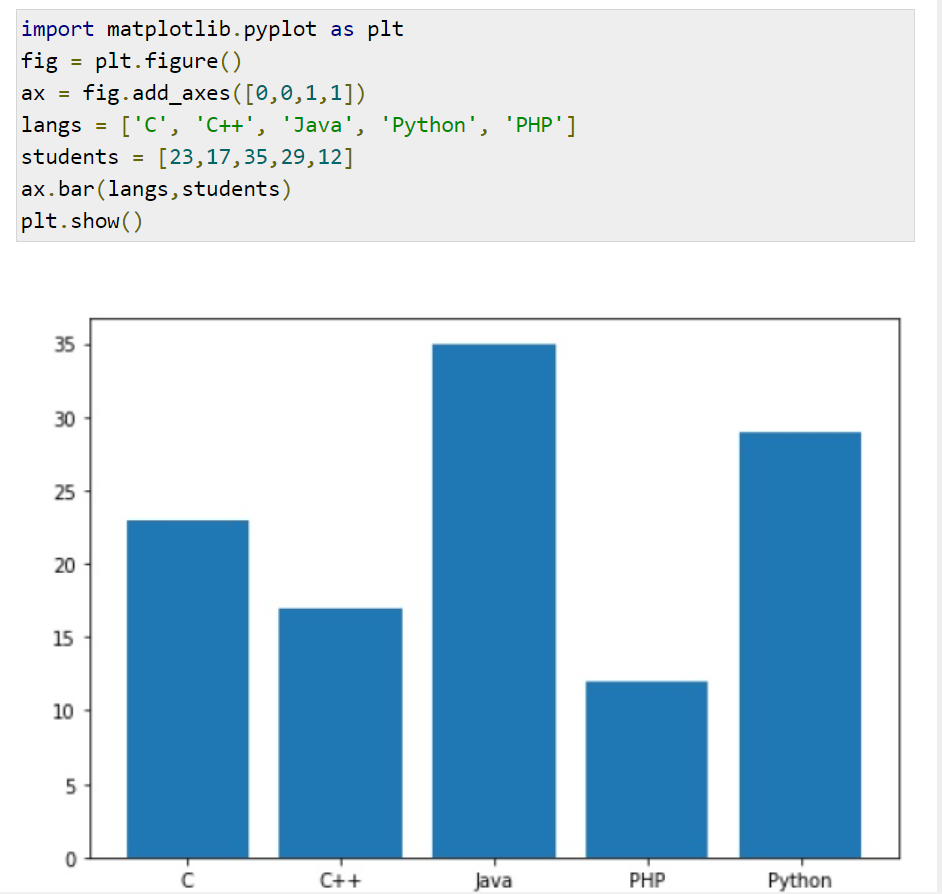
plt.show()

ax.bar(x, height, width, bottom, align)

The function makes a bar plot with the bound rectangle of size (x −width = 2; x + width=2; bottom; bottom + height).

The parameters to the function are −

|  |  |
| --- | --- |
| x | sequence of scalars representing the x coordinates of the bars. align controls if x is the bar center (default) or left edge. |
| height | scalar or sequence of scalars representing the height(s) of the bars. |
| width | scalar or array-like, optional. the width(s) of the bars default 0.8 |
| bottom | scalar or array-like, optional. the y coordinate(s) of the bars default None. |
| align | {‘center’, ‘edge’}, optional, default ‘center’ |



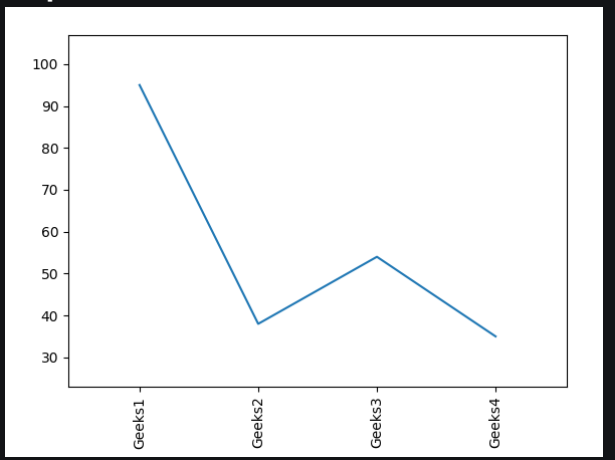
matplotlib.pyplot.xticks(ticks=None, labels=None, \*\*kwargs)

**Parameters:** This method accept the following parameters that are described below:

* **ticks:**This parameter is the list of xtick locations. and an optional parameter. If an empty list is passed as an argument then it will removes all xticks
* **labels:**This parameter contains labels to place at the given ticks locations. And it is an optional parameter.
* **\*\*kwargs:**This parameter is Text properties that is used to control the appearance of the labels.

example

# Implementation of matplotlib.pyplot.xticks()

# function

import numpy as np

import matplotlib.pyplot as plt

x = [1, 2, 3, 4]

y = [95, 38, 54, 35]

labels = ['Geeks1', 'Geeks2', 'Geeks3', 'Geeks4']

plt.plot(x, y)

# You can specify a rotation for the tick

# labels in degrees or with keywords.

plt.xticks(x, labels, rotation ='vertical')

# Pad margins so that markers don't get

# clipped by the axes

plt.margins(0.2)

# Tweak spacing to prevent clipping of tick-labels

plt.subplots\_adjust(bottom = 0.15)

plt.show()

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

table = pd.read\_csv("https://storage.googleapis.com/dqlab-dataset/penduduk\_gender\_head.csv")

table.head()

x\_label = table['NAMA KELURAHAN']

plt.bar(x=np.arange(len(x\_label)),height=table['LAKI-LAKI WNI'])

plt.xticks(np.arange(len(x\_label)), table['NAMA KELURAHAN'], rotation=30)

plt.show()

Memberi label pada grafik

plt.xlabel('Keluarahan di Jakarta pusat')

plt.ylabel('Jumlah Penduduk Laki - Laki')

plt.title('Persebaran Jumlah Penduduk Laki- Laki di Jakarta Pusat')

