Sharing Knowledge

Python The Series: Learning Python and Data Wrangling with Pandas

Open Source Community
Sunday, 6th October 2019
Bekasi

Asep Andri Fauzi

Introduce my self

Name	: Asep Andri Fauzi
Job	: Master’s Student, IPB University, Mayor Statistics
Address	: Dramaga, Kota Bogor
Activities	: Training Assistance in IYKRA Freelance Statistics Consultant Head of Scientific Division HIMPRO STK IPB
Email	: asepandrif@gmail.com

Disclaimer

- Seluruh sintaks ditulis agar dapat dieksekusi oleh Python 3
- Materi dalam slide ini hanya pemantik, banyak diskusi dilakukan secara langsung
- Materi dalam slide ini bersifat pengantar sehingga tidak sangat lengkap
- Beberapa sintaks direproduksi dari website resmi Python dan terinspirasi dari buku-buku tentang Python
- Penulisan baris kode dalam file ini tidak baik karena alasan teknis
- Tidak semua kodingan yang didiskusikan ditulis di sini

Objectives

Tujuan kita untuk hari ini: Pengantar

1. Mengenal Python dan lingkungannya
2. Mengerti beberapa bahasa Python dasar
3. Mengerti teknik-teknik manipulasi data (data wrangling) menggunakan Pandas

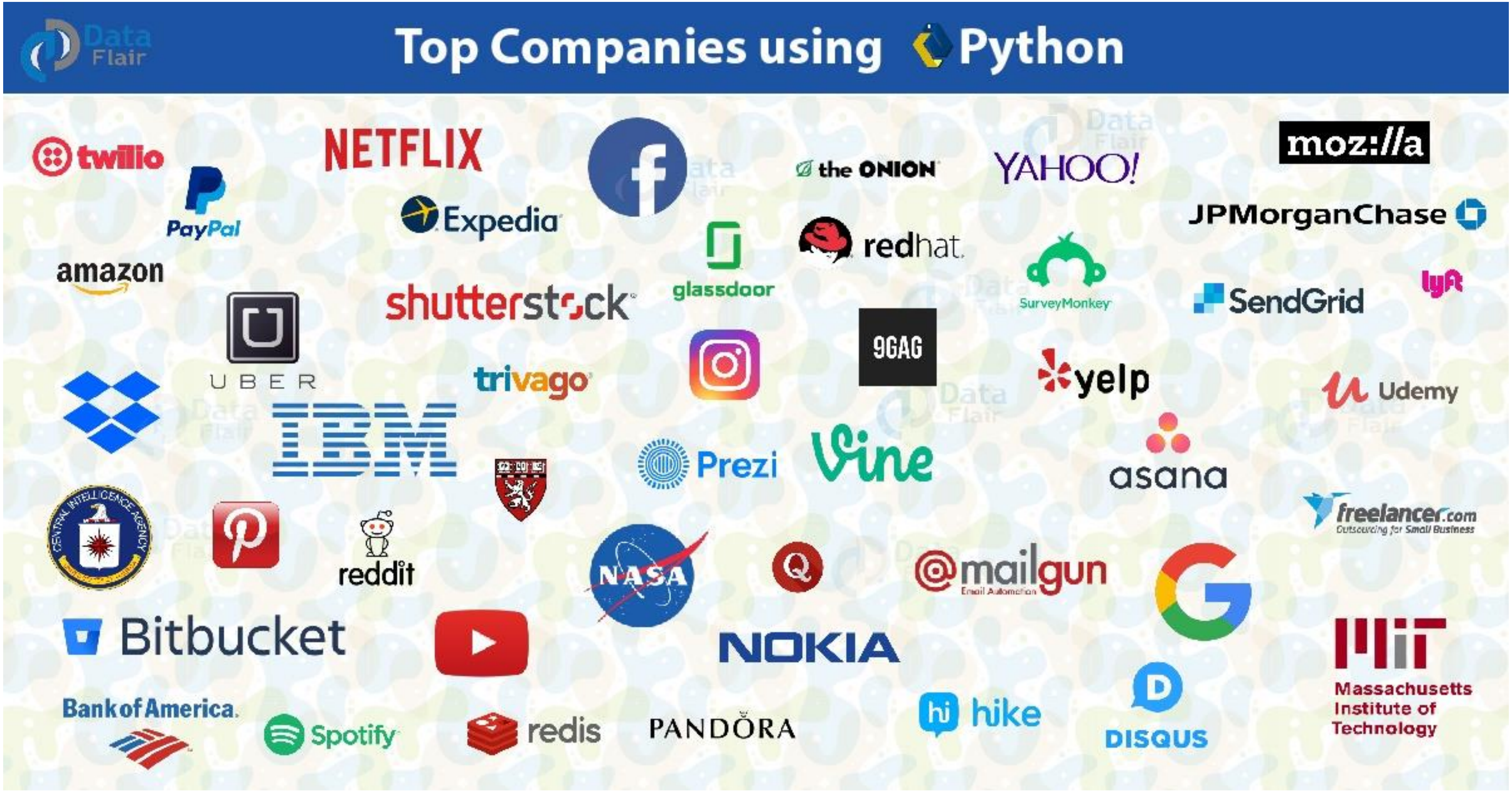
Metode belajar: Diskusi terpimpin

Semua orang boleh bertanya dan menjawab dalam topik yang diatur pemateri

Hasil:

Wawasan bertambah dan tergerak untuk *ngulik* lebih banyak

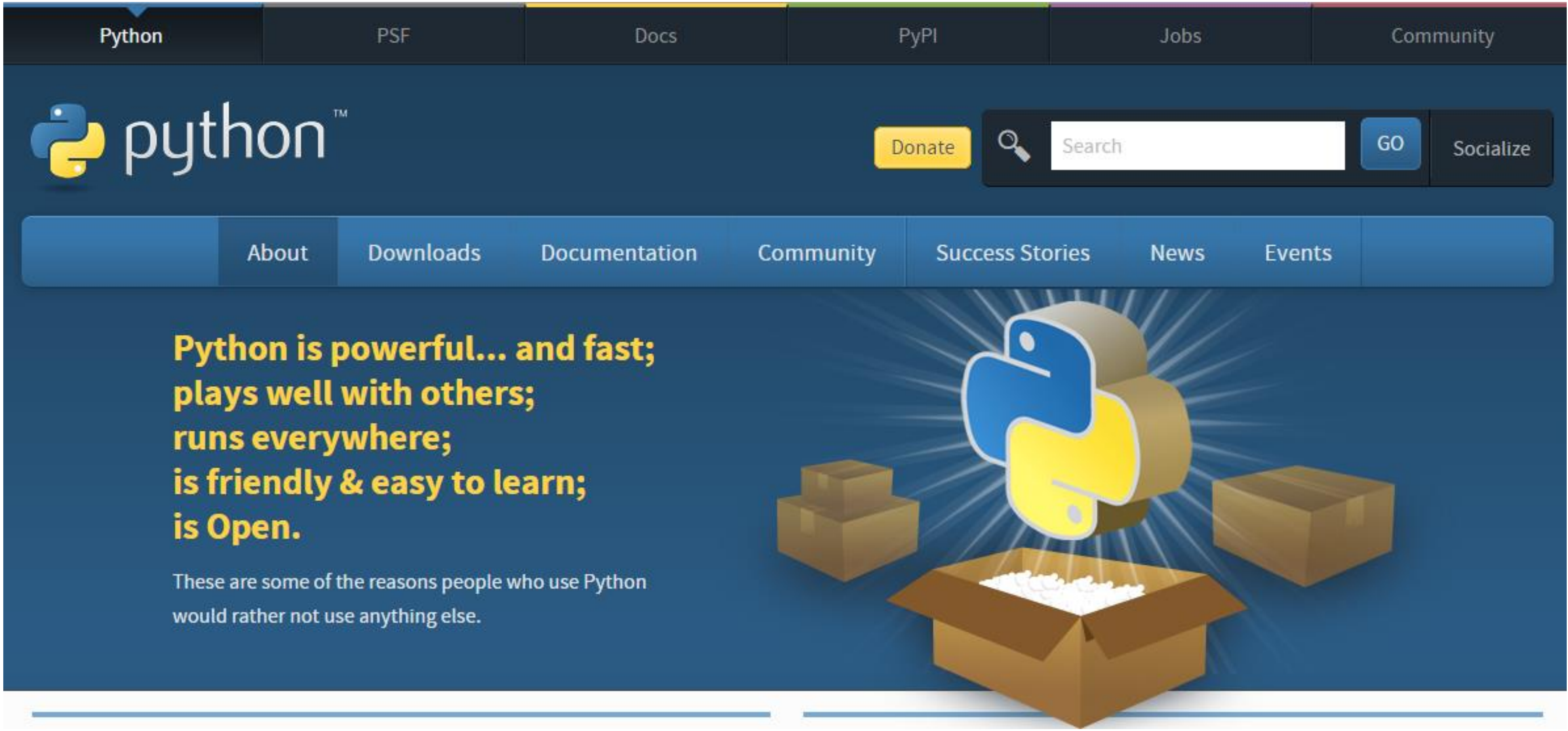
Which companies use Python?



Sumber: quora.com (2019)

Python's Official Website

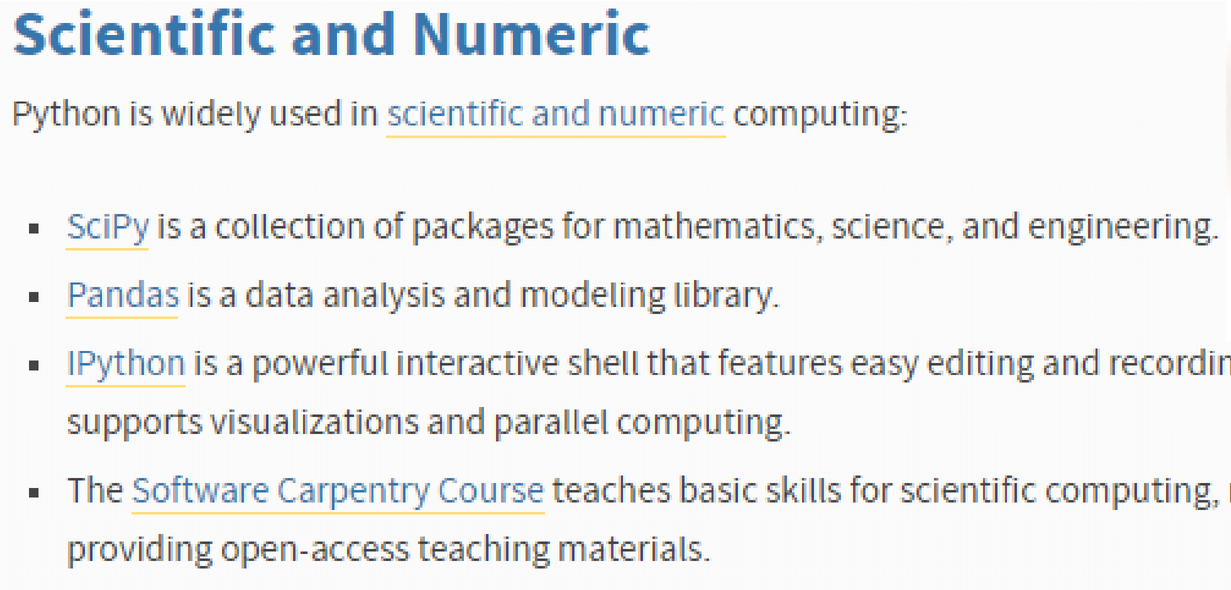
https://www.python.org/about/



Python is powerful... and fast;
plays well with others;
runs everywhere;
is friendly & easy to learn;
is Open.

These are some of the reasons people who use Python would rather not use anything else.

Applications



Scientific and Numeric

Python is widely used in scientific and numeric computing:

- SciPy is a collection of packages for mathematics, science, and engineering.
- Pandas is a data analysis and modeling library.
- IPython is a powerful interactive shell that features easy editing and recording of a work session, and supports visualizations and parallel computing.
- The Software Carpentry Course teaches basic skills for scientific computing, running bootcamps and providing open-access teaching materials.

Python is …

Programming language

Bahasa pemrograman yang mudah dibaca (readable) sehingga menjadi bersifat intuitif.

Interpreter

Setiap “baris” perintah akan langsung dieksekusi.

Open Source

Siapapun boleh berkontribusi. Lisensi Python:
PSF LICENSE AGREEMENT FOR PYTHON 3.7.5rc1

Python 2 ➤ Python 3

Terjadi beberapa perubahan.
Misalnya, **print objek** menjadi **print(objek)**

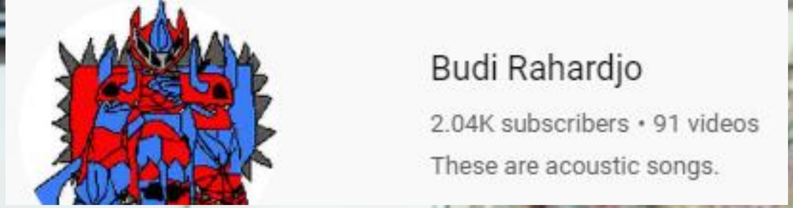
Video

Mari nonton video … :D

Sumber: TFiR

Kelahiran	31 January 1956
Kebangsaan	Belanda
Pendidikan	Magister Matematika dan Ilmu Komputer, University of Amsterdam (1982)
Pekerjaan	Google (2005 – 2012) Dropbox (2013 – Sekarang)
Tayangan favorit	Monty Python’s Flying Circus

YouTube



Budi Rahardjo

2.04K subscribers • 91 videos

These are acoustic songs.

Beware with …

Case Sensitive

A ≠ a

Shallow Copy

Saat kita menulis:

objek1 = objek2

maka kedua objek akan diletakan pada lokasi yang sama, sehingga perubahan pada salah satu objek dirasakan oleh objek lain.

Identation

Perintah di dalam looping dan pendefinisian fungsi ditulis menjorok ke dalam

Sumber referensi

Bedakan bahasa Python 2 dan 3

2020

Dukungan pada Python 2 akan dihentikan

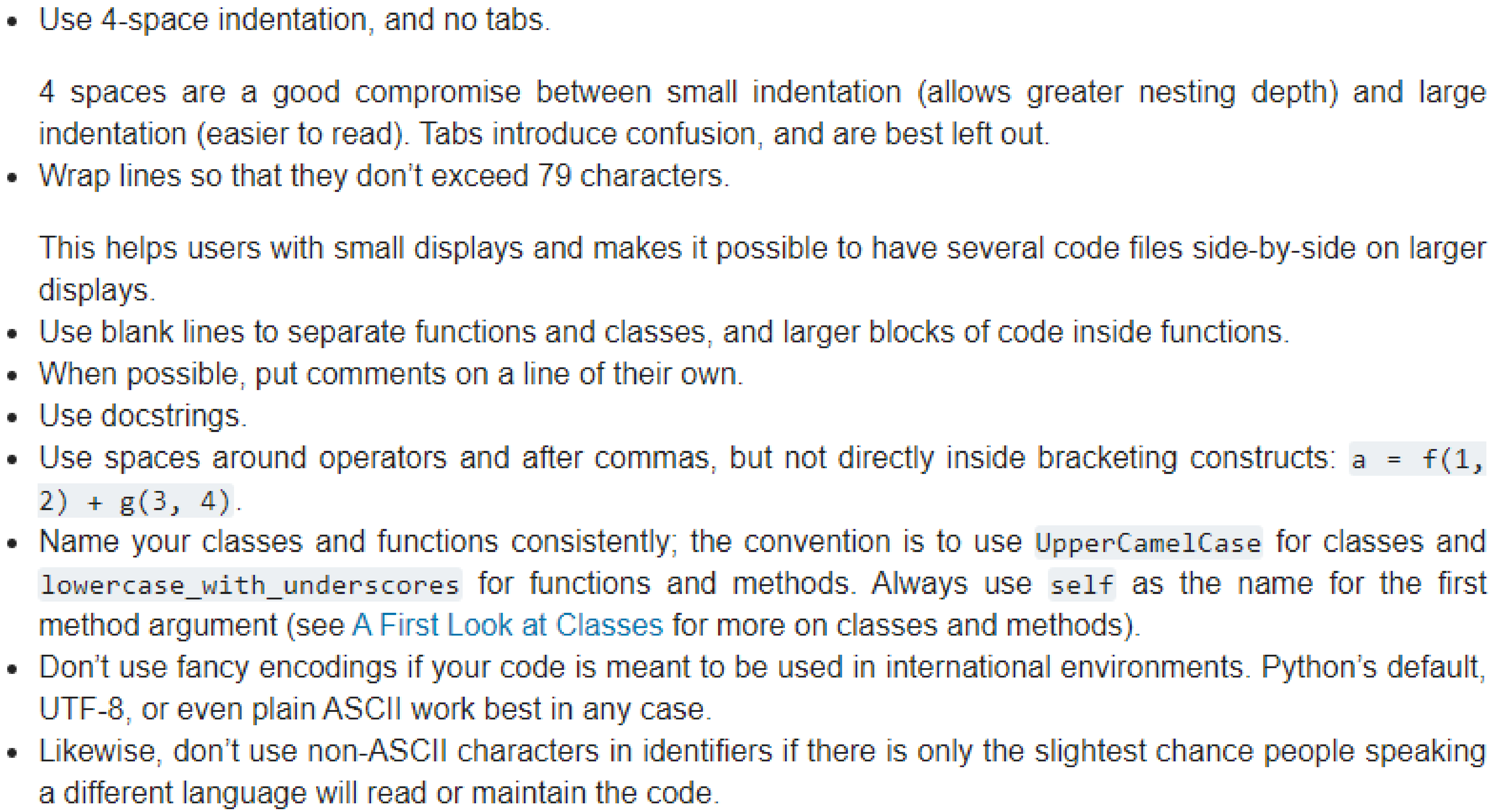
Let’s start

… … … Before we start, please let me ask:

is there any question?

Coding Style

For Python, **PEP 8** has emerged as the style guide that most projects adhere to; it promotes a very readable and eye-pleasing coding style.



- Use 4-space indentation, and no tabs.

  4 spaces are a good compromise between small indentation (allows greater nesting depth) and large indentation (easier to read). Tabs introduce confusion, and are best left out.
- Wrap lines so that they don't exceed 79 characters.

  This helps users with small displays and makes it possible to have several code files side-by-side on larger displays.
- Use blank lines to separate functions and classes, and larger blocks of code inside functions.
- When possible, put comments on a line of their own.
- Use docstrings.
- Use spaces around operators and after commas, but not directly inside bracketing constructs: `a = f(1, 2) + g(3, 4)`.
- Name your classes and functions consistently; the convention is to use `UpperCamelCase` for classes and `lowercase_with_underscores` for functions and methods. Always use `self` as the name for the first method argument (see A First Look at Classes for more on classes and methods).
- Don't use fancy encodings if your code is meant to be used in international environments. Python's default, UTF-8, or even plain ASCII work best in any case.
- Likewise, don't use non-ASCII characters in identifiers if there is only the slightest chance people speaking a different language will read or maintain the code.

PENGANTAR TIPE OBJEK DASAR

1 Numbers, Strings, and List

Python as a calculator (Numbers) & assignment

Mencetak (print) hasil

```
print(123) # fungsi print
123
```

Penjumlahan dan pengurangan

```
10 + 10 # penjumlahan
10 + -10 # penjumlahan dengan bilangan negatif
10 - 10 # pengurangan
```

Perkalian dan pembagian

```
6 * 3 # perkalian
6 * 1/3 # pekalian dengan hasil bagi
6/3 # pembagian
2 ** 3 # pangkat
25 ** (1/2) # akar pangkat dua
8 ** (1/3) # akar pangkat tiga
```

Modulus

```
7 % 2
11 % 3
```

Operator //

```
11 // 3
13 // 4
```

Penugasan (Assignment)

```
bonus
honor = 2000
pajak = 0.025
pendapatan = honor - honor * pajak
print(pendapatan)
```

Variabel _

```
total =  bonus + _
```

Working in strings

String pendek

```
print(‘pendek’)
print(“pendek”)
print(“don’t”)
```

Tanda \ (Menghilangkan pengaruh “ ”)

```
print(“don\'t”)
print(“\”Siap!\” kata orang itu”)
```

String panjang

```
Print(‘’’
string
panjang
‘’’)
```

Operator +

```
print(“Py” + “thon”)
prefiks = “di”
dasar_1 = “terima”
dasar_2 = “tolak”
print(prefiks + dasar_1)
print(prefiks, dasar_2)
print(“Py” “thon”)
```

Tanda \ dan r

Tanda \ berguna sebagai penginterpretasi karakter spesial sedangkan r membuat string menjadi terbaca sebagai *raw strings* (apa adanya)

```
print(“ganti baris \n ke bawah”)
print(“C:\invata\nama”) # \n turun satu baris
print(r“C:\invata\nama”)
```

Working in strings

Indexed

kata_sambutan = “Hello”

kata_sambutan[0]	indeks ke 0
kata_sambutan[:2]	indeks ke 0 sampai 2 – 1
kata_sambutan[1:2]	indeks ke 1 sampai 2 – 1
kata_sambutan[1:]	indeks ke 1 sampai terakhir
kata_sambutan[-1]	indeks ke 1 dari belakang
kata_sambutan[-0]	indeks ke 0
kata_sambutan[:-3]	dari indeks ke 0 sampai -3 – 1

H	e	l	l	o	← string
0	1	2	3	4	indeks
-5	-4	-3	-2	-1	

Metode

kata_sambutan.lower(); kata_sambutan.upper()

Tambahan infromasi

.lower()
.upper()
etc…

method

Fungsi len dan operator in

sambutan = kata_sambutan + “, Pak Febri. Ada yang bias Elen bantu?”
len(sambutan)
“Elen” in sambutan

Working in List

Create list

akurasi = [0.6,0.7, 0.8, 0.9]
kredit = [‘macet’, ‘lancar’, ‘macet’]
feature = [1,2,3,4]
container_1 = list()
container_2 = []
type(container_1) # mengecek tipe

Concatenating (operator +)

akurasi+feature
cacah = [1, 3, 5, 7, 9] + [2, 4, 6, 8]
dua_list = [akurasi, feature]
campur = feature + kredit
isistance(campur, str)

Concatenating (append & extend)

akurasi.append(1) # hanya 1 nilai
feature.extend([5,6,7])

Indexed

feature[0]
feature[:3]
feature[1:]
dua_list[0][2]

Fungsi len

len(feature)
len(dua_list)

More on List

akurasi.insert(0,0.2)
akurasi.pop() # menghapus nilai yang ditambahkan
del campur [0] # menghapus nilai pada indeks ke 0
kredit.reverse() # membalik urutan nilai
kredit.count(‘macet’) # menghitung nilai ‘macet’
kredit.index(‘macet) # mencari indeks nilai ‘macet’
kredit.sort() # mengurutkan data secara, ascending (reverse = F)

Edited

cacah[1] = 3
cacah[1:4] = [9,9,9]

Tambahan infromasi

.append()
.count()
etc…

method

STRUKTUR DATA LAINNYA

2 Range, Tuple, Set, Frozenset dan Dictionary

Range, Tuple

Tuple is **immutable**

Create Range

```
r_1 = range(10)
r_2 = range(1,10,2)
r_3 = range(10,0,-1)
```

Create Tuple

```
t_1 = 12345, 54321, 'hello!‘
t_2 = (1,’a’,2,’b’)
t_3 = tuple([1,2,3])
t_4 = tuple(r_2)
```

Indexed

```
t[0]
```

sama seperti List

Tuples may be nested

```
u = t, (1, 2, 3, 4, 5)
```

Tuples are immutable

```
t[0] = 88888 # error
```

But they can contain mutable objects

```
v = ([1, 2, 3], [3, 2, 1])
```

More in Tuple

```
empty = ()
singleton = 'hello',    # note trailing comma
len(empty)
len(singleton)
x, y, z = t # sequence unpacking
x; y; z
```

Set, Frozenset, Dictionary

Frozenset seperti set tapi elemennya tidak bisa ditambah atau dikurangi

Create Set

```
basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
basket = set('apple', 'orange', 'apple', 'pear', 'orange', 'banana')
```

Method: add dan remove

```
basket.add(‘belimbing’)
basket.remove(‘belimbing’)
```

Operator in

```
'orange' in basket
'crabgrass' in basket
```

Set operations on unique letters from two words

```
a = set('abracadabra')
b = set('alacazam')
a # unique letters in a
a - b # letters in a but not in b
a | b # letters in a or b or both
a & b # letters in both a and b
a ^ b  # letters in a or b but not both
```

Create Set

```
himp_beku = frozenset(['januari', 'februari', 'maret'])
```

Dictionary

```
tel = {'jack': 4098, 'sape': 4139}
tel['guido'] = 4127 # insert guido
tel['jack'] # print jack’s value
```

Delete and insert in dictionary

```
del tel['sape']
tel['irv'] = 4127
```

Convert to list

```
list(tel)
sorted(tel)
```

Operator in

```
'guido' in tel
'jack' not in tel
```

Control Flow and Function

3 If, for, and while

Function

if Statements

if

```
x = int(input("Please enter an integer: "))

if x < 0:
    x = 0
    print('Negative changed to zero')
elif x == 0:
    print('Zero')
elif x == 1:
    print('Single')
else:
    print('More')
```

pass Statment

```
xx = 1

if xx % 2 == 0:
    print(‘genap’)
else:
    pass # do nothing
```

for, while Statements

For

```
words = ['cat', 'window', 'defenestrate']
for w in words:
    print(w, len(w))

for w in words[:]:
    if len(w) > 6:
        words.insert(0, w)

bilangan = [11, 12, 13]
for i, nilai in enumerate(bilangan, 1): # start indeks from 1
    print(i, nilai, nilai**2)
```

While

```
count = 0
while (count < 9):
   print('The count is:', count)
   count = count + 1
```

Break Statement

```
for n in range(2, 10):
    for x in range(2, n):
        if n % x == 0:
            print(n, 'equals', x, '*', n//x)
            break
    else:
        # loop fell through without finding a factor
        print(n, 'is a prime number')
```

Continue Statement

```
for num in range(2, 10):
    if num % 2 == 0:
        print("Found an even number", num)
        continue
    print("Found a number", num)
```

Function

Just Print

```
def fib(n):    # write Fibonacci series up to n
    """Print a Fibonacci series up to n."""
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()

fib(2000) # call the function
```

Print with Special Character %s, %d, %.f

```
fibo = “Fibonacci”
suku = 1
def KarakterSpesial():
    print(“Deret %s adalah suku ke %d adalah 
%.1f” % (fibo, suku, f100[suku-1]))
```

Store Result

```
def fib2(n):  # return Fibonacci series up to n
    """Return a list containing the Fibonacci series up to n."""
    result = []
    a, b = 0, 1
    while a < n:
        result.append(a)    # see below
        a, b = b, a+b
    return result

f100 = fib2(100)    # call it
f100                # write the result
```

Fungsi lambda

lambda seperti def dengan return

```
sqrt = lambda x: x**(1/2) # membuat fungsi akar
```

Python Packages

Packages, modules

In addition to the standard library, there is a growing collection of several thousand components (from individual programs and modules to packages and entire application development frameworks), available from the Python Package Index. The following table contains some packages in python.

Module	Keterangan	Function
Built in	Tidak perlu dipanggil	abs(), dict(), dir(), enumerate(), frozenset(), input(), int(), list(), max(), min(), object(), range(), round(), set(), sum(), tuple(), type()
math	Matematika	ceil(), factorial(), fsum(), isfinite(), isinf(), isnan(), exp(), log(), log10(), sqrt()
statistics	Statistika	mean(), harmonic_mean(), median(), mode(), pstdev(), pvariance(), stdev(), variance()
copy	Menyalin data	copy(), deepcopy()
pickle	Implementasi	dumps(), loads()
sqlite3	Database	commit(), rollback(), execute(), keys()
numpy	Array computing	array(), linspace(), arrange(), mean(), median(), nanmean(), nanmedian()
pandas	Data manipulation	Series(), DataFrame(), dypes(), index(), columns(), dropna(), fillna(), stack(), aggregate(), groupby, count()
scikit-learn	Machine learning	…

Pandas
An Introduction

History of Development

In 2008, *pandas* development began at AQR Capital Management. By the end of 2009 it had been open sourced, and is actively supported today by a community of like-minded individuals around the world who contribute their valuable time and energy to help make open source *pandas* possible. Thank you to all of our contributors.

Since 2015, *pandas* is a NumFOCUS sponsored project. This will help ensure the success of development of *pandas* as a world-class open-source project.

Pandas adalah pustaka Python yang menyajikan struktur data yang cepat, fleksibel, dan ekspresif dengan tujuan untuk membuat pengolahan data lebih mudah dan intuitif. Pandas dapat menampung tipe data yang heterogen. Tulisan ini hanya berisi sedikit saja tentang pandas. Adapun resep atau tutorial lengkap tentang pandas sudah disediakan oleh pengembang.

Ekosistem pandas sangat besar dan terbuka. Issue tentang pandas dapat dilihat di github pandas dan jika berhadapan dengan persoalan yang belum dapat diatasi kita dapat bertanya (dengan kemungkinan terjawab sangat besar) di stackoverflow. Pengembangan pandas dilakukan oleh banyak orang tapi orang-orang yang tergabung dalam tim core pandas dapat kita lihat di sini. Selain individu, pandas juga disokong oleh institusi Anaconda, Two sigma, dan Paris-Saclay,

Import library

```
import numpy as np
import pandas as pd
```

Pandas dtype

Pandas dtypes mapping

Pandas dtype	Python type	NumPy type	Usage
object	str	string_, unicode_	Text
int64	int	int_, int8, int16, int32, int64, uint8, uint16, uint32, uint64	Integer numbers
float64	float	float_, float16, float32, float64	Floating point numbers
bool	bool	bool_	True/False values
datetime64	NA	datetime64[ns]	Date and time values
timedelta[ns]	NA	NA	Differences between two datetimes
category	NA	NA	Finite list of text values

Sumber: pbpython.com

Pandas Series

Create Pandas Series

```
ss = pd.Series([1, 3, 5, np.nan, 6, 8])
type(ss)
```

Index berupa tanggal

```
tanggal = pd.date_range('20181222', periods = 10)
dt = pd.Series(np.random.randn(10), index = tanggal)
```

Index berupa string

```
dts = pd.Series(np.random.randn(3), index = ['asep', 'andri', 'fauzi'])
```

Index tidak beraturan

```
dn1 = pd.Series(np.random.randn(3), index = [10, 4, 12])
```

Memuat tipe data campuran

```
dc1 = pd.Series([1,2,'tiga',4, 0.4])
```

Melihat isi Pandas Series

```
ss.values # mengakses nilai ss
ss.dtypes # mengakses tipe data ss
dc1.dtypes
```

Pandas DataFrame

DataFrame vs Series

```
df = pd.DataFrame(ss) # perhatikan tampilannya
```

Mengecek ukuran objek

```
import sys
print("Ukuran series ss adalah", ss.nbytes, "bytes")
print("Ukuran dataframe df adalah", sys.getsizeof(df),
"bytes")
```

Create DataFrame

Mengubah dimensi list

```
series = list(range(0,99,10))
df_list = pd.DataFrame(np.reshape(series,(5,2)),
columns = ["x1", "x2"])
```

Membuat dari kamus

```
kamus = {'benua': ('asia', 'eropa', 'amerika'),
        'negara': ('jepang','inggris', 'canada'),
        'jumlah': (200, 203, 179)}
df_kamus = pd.DataFrame(kamus)
```

Mengubah dimensi Pandas Series

```
df_series = pd.DataFrame(ss.values.reshape(2,3), columns
= ["s1", "s2", "s3"])
```

Membuat DataFrame dengan index tertentu

```
tanggal
nilai = np.random.binomial(n=1, p=0.5, size = 20)
df_indeks = pd.DataFrame(nilai.reshape(10,2), index =
tanggal, columns = ['kolom 1', 'kolom 2'])
```

Melihat “isi” DataFrame

```
df_info = pd.DataFrame({'A': 1.,
                        'B1': pd.Timestamp('20130102'),
                        'B2': pd.date_range("09:00", "10:30", freq="30min").time,
                        'C': pd.Series(1, index=list(range(4)), dtype='float32'),
                        'D': np.array([3] * 4, dtype='int32'),
                        'E': pd.Categorical(["d", "e", "c", "b"]),
                        'F': 'foo'})

df_info.info() # informasi penting
df_info.dtypes # type data perkolom

df_large = pd.DataFrame(np.random.randn(1000,5), # membangkitkan bilangan acak normal baku ukuran 1000 x 5
                        index = pd.date_range('20000101', periods = 1000),
                        columns= ['x1','x2','x3','x4','x5'])
print('Dimensi DataFrame df_large adalah', df_large.shape)
df_large.head(3) # 3 baris pertama
df_large.tail(4) # 4 baris terakhir
df_large.index # nama index
df_large.columns # nama kolom
```

Summary statistic, DataFrame to numpy

Melihat ringkasan statistik

```
df_large.describe()
df_info.describe()
df_info[['B1','B2','E']].describe()
```

Convert DataFrame to numpy

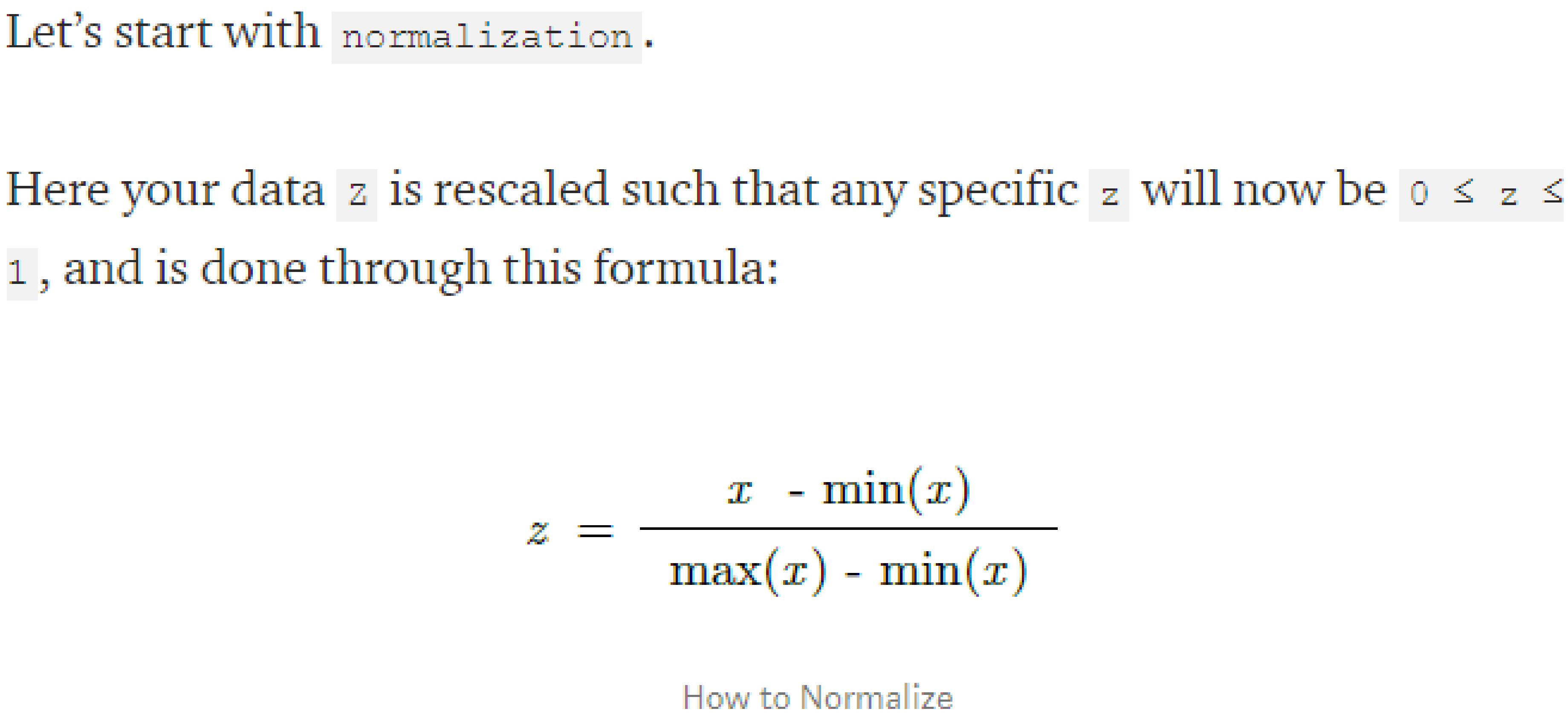
```
numpy_df = df_info.to_numpy()
```

Practice case: HR Analytics

Dataset description

Variable	Definition
employee_id	Unique ID for employee
department	Department of employee
region	Region of employment (unordered)
education	Education Level
gender	Gender of Employee
recruitment_channel	Channel of recruitment for employee
no_of_trainings	no of other trainings completed in previous year on soft skills, technical skills etc.
age	Age of Employee
previous_year_rating	Employee Rating for the previous year
length_of_service	Length of service in years
KPIs_met >80%	if Percent of KPIs(Key performance Indicators) >80% then 1 else 0
awards_won?	if awards won during previous year then 1 else 0
avg_training_score	Average score in current training evaluations
is_promoted	(Target) Recommended for promotion

Normalisasi



Let’s start with `normalization`.

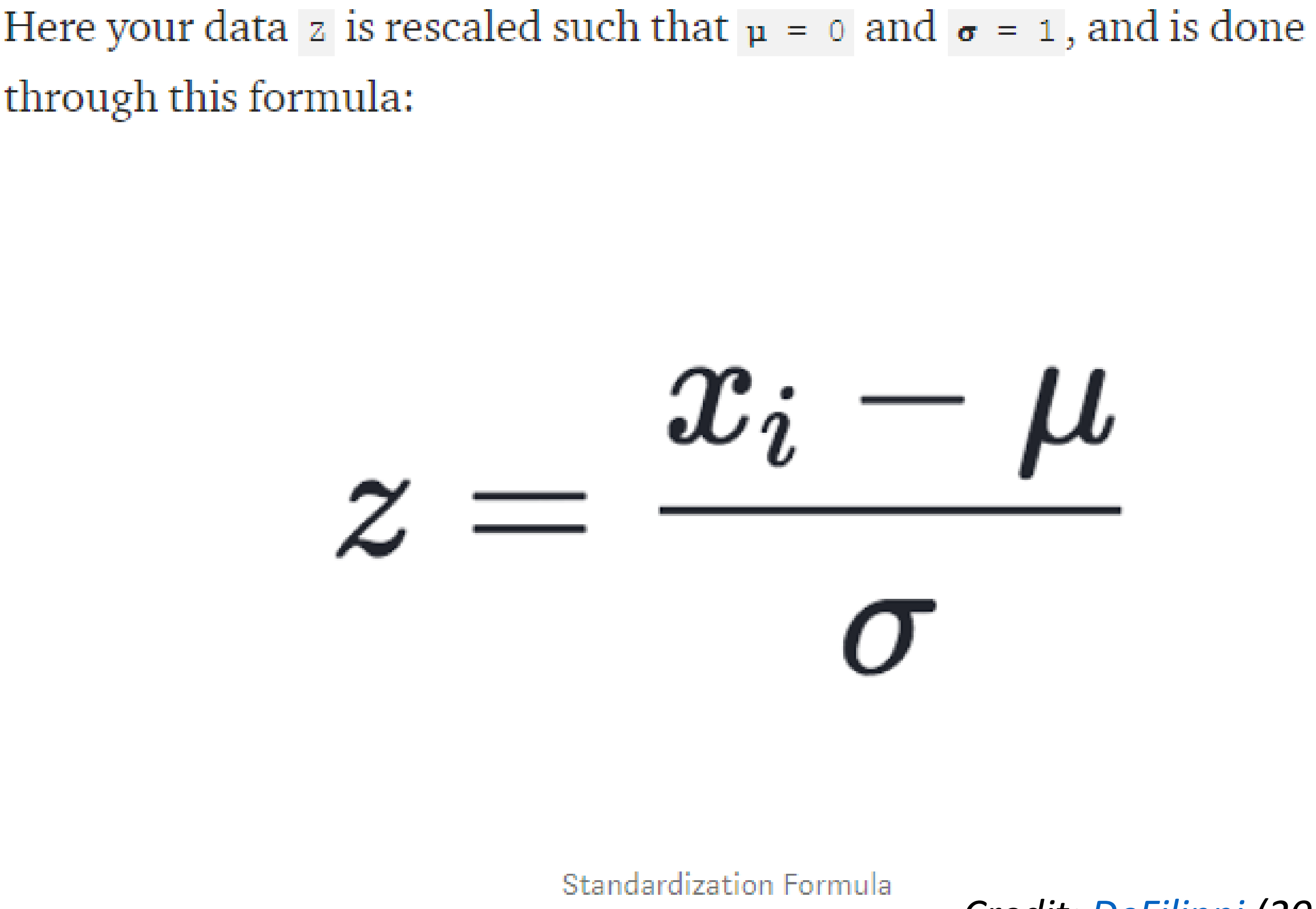
Here your data `z` is rescaled such that any specific `z` will now be `0 ≤ z ≤ 1`, and is done through this formula:

$$z = \frac{x \ - \min(x)}{\max(x) - \min(x)}$$

How to Normalize

Credit: *DeFilippi (2018)*

Standarisasi

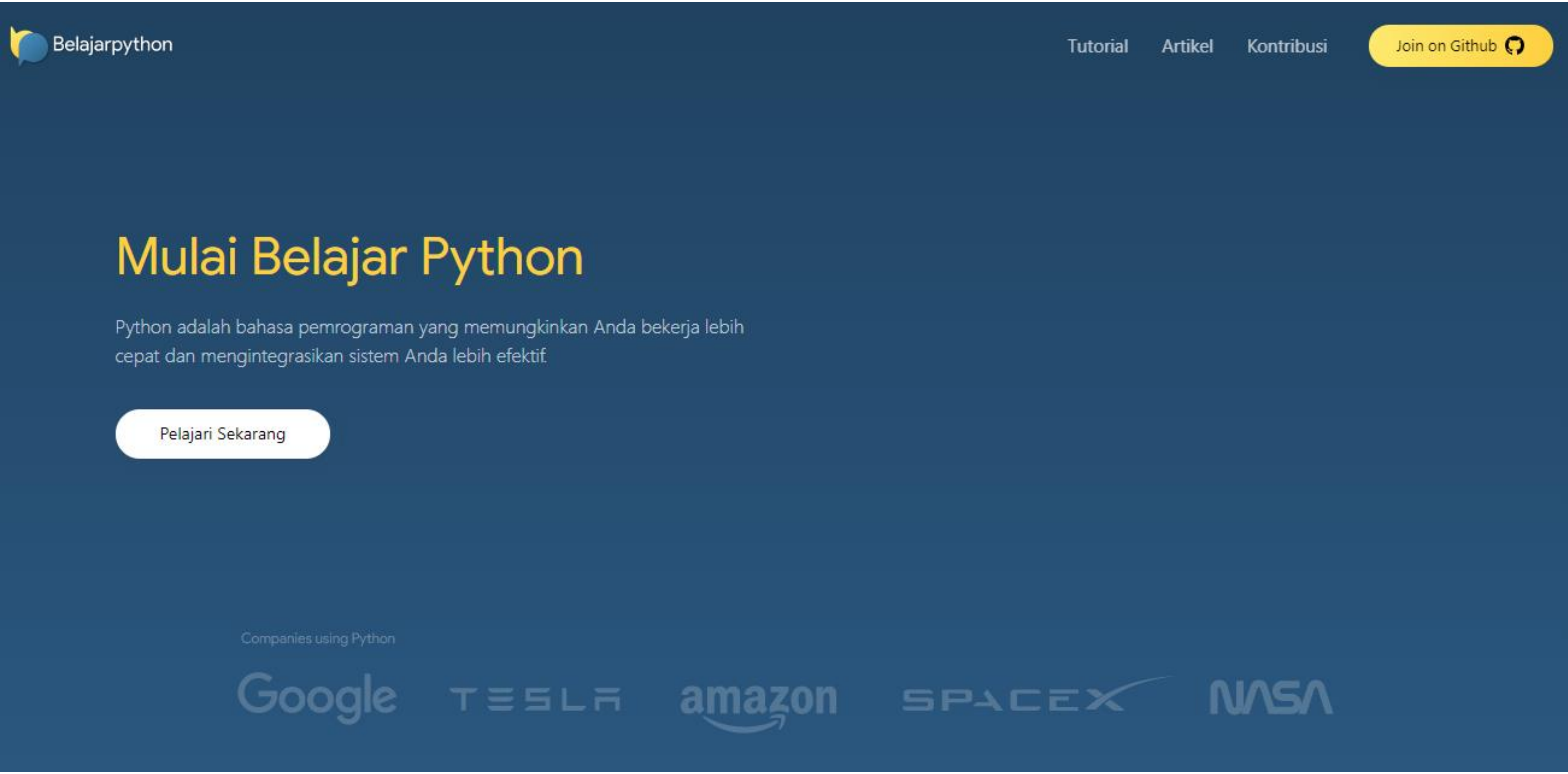


Here your data `z` is rescaled such that `μ = 0` and `σ = 1`, and is done through this formula:

$$z = \frac{x_i - \mu}{\sigma}$$

Standardization Formula

Credit: DeFilippi (2018)



Mulai Belajar Python

Python adalah bahasa pemrograman yang memungkinkan Anda bekerja lebih cepat dan mengintegrasikan sistem Anda lebih efektif.