**Day 1 – Setting Environment Python-Data Science**

Assalamu’alaikum warahmatullahi wabarakatuh..

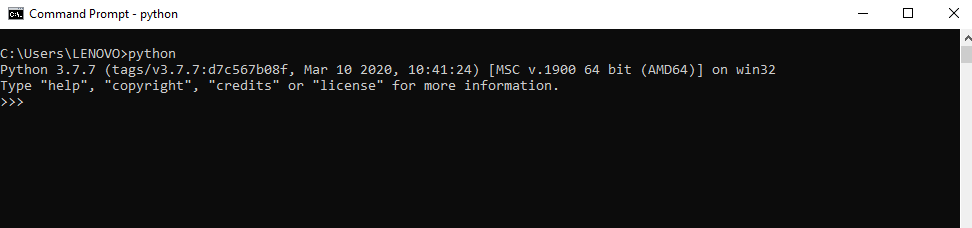
Artikel ini merupakan bab awal yang mencoba untuk membuat kalian familiar dengan python agar kalian siap untuk mempelajari data science.

Ok, sebelum kita memulai course ini, pertama2 kita akan mempersiapkan environment kita sehingga kita mempunyai tool yang cukup untuk mempelajari data science. Tool yang akan kita butuhkan diantaranya :

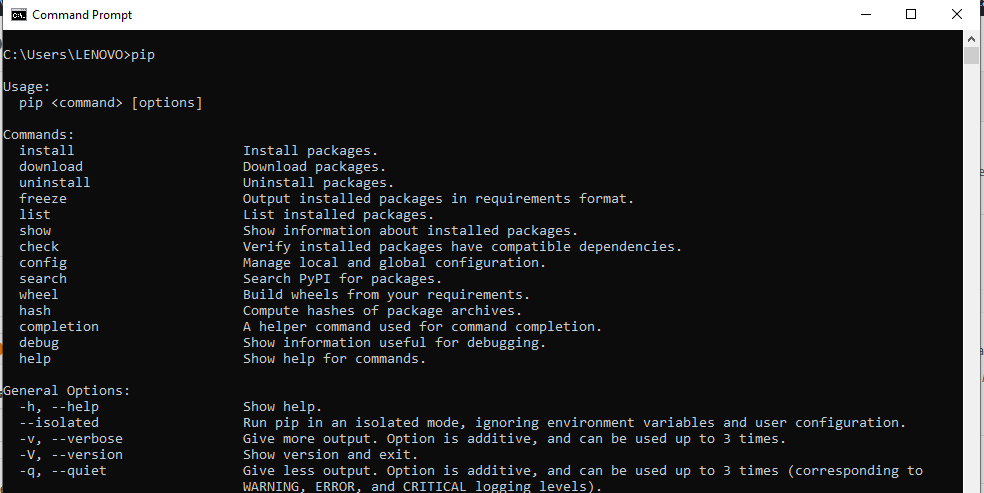
1. Install Python
2. Pip ( Python Pacakage Manager )
3. Install beberapa library/module agar bisa menjalankan jupyter notebook

Pertama, untuk install python, kalian bisa download installernya disini [python.org](https://www.python.org/). Kalian pilih installer yang sesuai dengan operating system kalian. **Catatan:** saat ini ada 2 versi python, yaitu python 2.7 dan 3.x, python versi 2.7 pengembangannya sudah dihentikan pada saat artikel ini ditulis. Maka kalian dianjurkan untuk menginstall python > 3.7.

Setalah kalian menginstall python, kalian dapat memastikannya dengan membuka commandprompt kemudian ketikan saja “python”. Setelah itu enter kemudian kalian akan masuk ke environment python.

cek python sudah terinstall

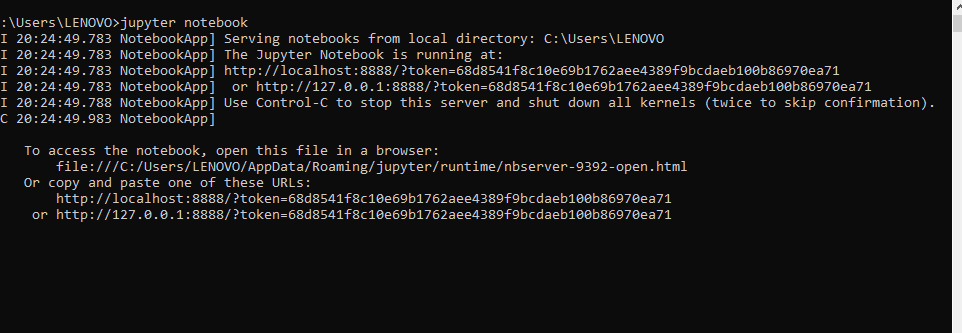
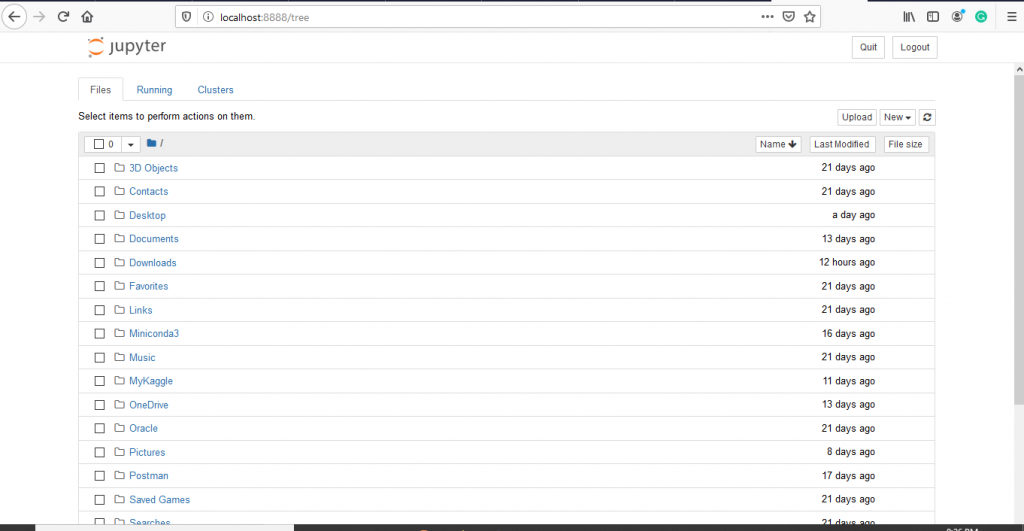
Pip ( python package manager ) juga sudah terinstall di komputer/laptop kalian. Kalian dapat memastikannya dengan ketik pip kemudian enter maka akan muncul tampilan sebagai berikut :

check pip sudah terinstall

Terakhir, untuk mempersiapkan environment, kalian perlu menginstall beberapa library untuk menjalankan jupyter notebook. Untuk menginstallnya, kalian buka command prompt kemudian lakukan perintah berikut :

***pip install ipython notebook***

Setelah selesai process installasi, kalian bisa check kalau kita telah bisa menjalankan jupyter notebook dengan mengetikan *jupyter notebook* di cmd, kemudian enter. Tunggu beberapa saat, nanti aplikasi jupyter notebook akan terbuka di default browser kalian dengan sendirinya. Disarankan untuk tidak dibuka di internet exploler.

membuka applikasi jupyter notebookTampilan awal jupyter notebook

Mulai saat ini, kita akan belajar python + data science di jupyter notebook ini. Jupyter notebook adalah suatu aplikasi dimana kita bisa membuat kode dan menjalankannya secara interaktif. Untuk lebih jauhnya, kalian bisa mempelajari apa itu jupyter notebook [disini.](https://indoml.com/2019/09/29/pengenalan-dan-panduan-jupyter-notebook-untuk-pemula/) => <https://indoml.com/2019/09/29/pengenalan-dan-panduan-jupyter-notebook-untuk-pemula/>

NOTE pada saat install ipython notebook : WARNING: You are using pip version 19.2.3, however version 20.0.2 is available. You should consider upgrading via the 'python -m pip install --upgrade pip' command.

# Terlindungi: Day 1 – Hello Python

Assalamu’alaikum warahmatullahi wabarakatuh..

Python adalah salah satu bahasa pemrograman yang sedang naik daun. Kepopulerannya terus meroket sejak tahun 2010. Python menjadi bahasa pemrograman yang mudah untuk dipelajari, termasuk bagi orang-orang yang baru belajar bahasa pemrograman.

Kemudian apa yang membuat python begitu populer? Beberapa diantaranya adalah :

1. Python adalah general multipurpose programming. Artinya, dengan bahasa pemrograman python, kita hampir bisa membuat berbagai macam aplikasi, contohnya web, desktop, game, dan lain sebagainya.
2. Python mudah untuk dipelajari. Python didesain dengan penekanan agar mudah untuk dibaca. Kita bisa menuangkan konsep aplikasi kita kedalam bahasa python dengan baris yang relatif sedikit, sehingga kita bisa lebih cepat dan efektif dalam membangun suatu aplikasi.
3. Python open source. Artinya, python gratis dan setiap orang bisa berkontribusi untuk perkembangan python dan membagikannya ke orang lain.

Untuk memahami lebih lanjut, silahkan kunjungi external resource yang telah kami sediakan di link di bawah ini :

## External Source

List Artikel :

* <https://www.petanikode.com/python-pip/>
* <https://www.petanikode.com/python-sintaks/>
* <https://www.petanikode.com/python-variabel-dan-tipe-data/>
* <https://www.petanikode.com/python-operator/>

List Video :

* <https://www.youtube.com/watch?v=SsUF-7X3ehM&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1> ( Pengenalan Python )
* <https://www.youtube.com/watch?v=rqt-T1kl4PU&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=3> ( Number dan Operasi Matematika )
* <https://www.youtube.com/watch?v=xlWN2fTyQFU&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=4> ( Python String )

Updated on April 20, 2020

# Day 2 – Python List

List adalah salah satu tipe data untuk mengoleksikan data di python. Contohnya, kita punya koleksi data suhu semua anggota keluarga kita, untuk menyimpan data ini, tentunya lebih mudah kalau kita menyimpannya dalam satu tempat, dan tempat itu salah satunya bisa bernama list. Mengapa salah satunya? Karena tempat menyimpan koleksi data di python tidak hanya list, ada tipe data lain seperti dictionary, tuple, set dan lain-lain. Namun dalam pembahasan ini kita akan fokus membahas tentang list.

Sampai di sini setidaknya kita sudah dapat sedikit gambaran tentang apa itu list. Selanjutnya kita akan coba pelajari bagaimana cara membuat list di python.. Berikut caranya..

# inisialisasi suhu anggota keluarga

suhu1 = 21

suhu2 = 20

suhu3 = 22

# simpan dalam list

suhu\_keluarga = [suhu1, suhu2, suhu3]

## Akses Data List

Setelah kita bisa membuat data sederhana dengan list. Sekarang kita akan belajar bagaimana caranya mengakses data dalam list. Untuk mengakses data dalam list, python menggunakan sesuatu yang bernama index. Index menunjukan posisi suatu data di dalam list, dan python memulai index dari 0. Perlu diketahui sebelumnya bahwa ada 2 teknik untuk mengakses data di dalam list. Pertama dengan subsetting list, kedua dengan slicing list. Mari kita lihat contohnya :

# Membuat data list

>>> tinggi\_badan = [

162, # index 0

177, # index 1

182, # index 2

150, # index 3

166 # index 4

]

# Subsetting list

>>> print(tinggi\_badan[0]) # posisi pertama

162

>>> print(tinggi\_badan[-2]) # posis kedua dari belakang

150

# Slicinglist

# Mengambil data dengan index 0, 1, 2, 3

>>> print(tinggi\_badan[:4])

[162, 177, 182, 150, 166]

# Mengambil data dengan index 2, 3, 4

>>> print(tinggi\_badan[2:5])

[182, 150, 166]

## Sifat List

Sebelumnya, kita sudah mengenal sedikit tentang list, mari kita kenal lebih dalam lagi tentang list.

1. List berisi koleksi nilai/data.
2. List bisa berisi tipe data apapun dan tidak harus semua data berisi tipe data yang sama.
3. List bisa berubah

Untuk sifat no 1. sudah dijelaskan dan tunjukan di atas. Sekarang kita langsung bahas sifat no.2. Sebelumnya, kita sudah mengenal tentang beberapa tipe data dalam list, ada numerik, string, boolean, dan sebagainya. Nah, nilai-nilai yang bertipe data apapun dapat dimasukan sebagai koleksi di dalam list, bahkan list itu sendiri bisa ada di dalam list. Untuk lebih jelasnya mari kita lihat contoh berikut..

# membuat list

suhu\_keluarga\_ucup = [

'ayah ucup', 19, 'ucup', 19, 'ibu ucup', 20

]

suhu\_keluarga\_boy = [

'istri boy', 20, 'anak boy', 18, 'istri kedua boy', 21

]

# membuat list dalam list dan di campur dengan data boolean

suhu\_keluarga = [

suhu\_keluarga\_ucup, suhu\_keluarga\_boy

]

## Manipulasi List

List adalah “mutable”, artinya dapat diubah.  
  
Salah satu cara untuk mengubah daftar adalah dengan menetapkan indeks atau ekspresi irisan.  
#  
Misalnya, katakanlah kita ingin mengubah suhu ucup:

# mengubah suhu ucup

suhu\_keluarga\_ucup[3] = 22

print(suhu\_keluarga\_ucup)

>>> ['ayah ucup', 19, 'ucup', 22, 'ibu ucup', 20]

katakanlah kita ingin mengganti ‘ibu ucup’ dengan ‘mamah ucup’, beserta suhunya :

suhu\_keluarga\_ucup[-2:] = ['mamah ucup', 22]

print(suhu\_keluarga\_ucup)

>>> ['ayah ucup', 19, 'ucup', 22, 'mamah ucup', 22]

kita juga bisa menambahkan elemen di list, yaitu dengan menggunakan ‘+’ operator, katakanlah kita ingin menambahkan adik ucup beserta suhunya, berikut contohnya :

suhu\_keluarga\_ucup = suhu\_keluarga\_ucup + ['adik ucup', 20]

print(suhu\_keluarga\_ucup)

>> ['ayah ucup', 19, 'ucup', 22, 'mamah ucup', 22, 'adik ucup', 20]

terakhir, kita juga bisa menhilangkan element di dalam list, yaitu seperti ini :

del(suhu\_keluarga\_ucup[0])

Perhatian, setelah suatu elemen di dalam list dihapus, maka index dari seluruh elemennya pun akan berubah, contohnya di atas kita mendelete elemen dengan index 0 di list, artinya kita mendelete elemen ‘ayah ucup’, maka elemen yang lain akan berubah menyesuaikan dengan perubahan tersebut, maka ketika kita mengambil index 0 di list tersebut maka hasilnya akan seperti ini :

print(suhu\_keluarga\_ucup[0])

>>>19

## Some Function in List

Python memiliki beberapa fungsi bawaan yang berguna untuk bekerja dengan list. diantaranya adalah :

‘len’ memberikan panjang daftar:

keluarga\_ucup = ['mamah', 'papah', 'ucup', 'adek ucup', 'kaka ucup']

# berapa banyak anggota keluarga ucup

print(len(keluarga\_ucup))

>>>5

‘sorted’ mengurutkan elemen list:

>>>print(sorted(keluarga\_ucup))

>>>['adek ucup', 'kaka ucup', 'mamah', 'papah', 'ucup']

‘sum’, tentu fungsi ini untuk menjumlahkan:

primes = [2, 3, 5, 7]

print(sum(primes))

>>>17

## External Resource

List Artikel/Tutorial :

* <https://www.petanikode.com/python-list/>

List Video :

* <https://www.youtube.com/watch?v=kuLBqkpnKDk&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=5> (Python List)
* <https://www.youtube.com/watch?v=NoX5wnjBaHo&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=17> (Python List 2)

# Terlindungi: Day 3 – Functions, Method and Getting Help

Sebelumnya kita telah melihat dan menggunakan fungsi-fungsi seperti print, sum, len dan sebagainya. Python memiliki lebih banyak fungsi, dan membuat fungsi kita sendiri adalah bagian besar dari perjalanan kita belajar pemrograman python.  
  
Dalam bagian ini kita akan belajar lebih banyak tentang menggunakan dan mendefinisikan fungsi.

## Getting Help

Kita melihat fungsi ‘len’ di tutorial sebelumnya, tetapi bagaimana jika kita lupa fungsi apa itu?  
  
Fungsi help () merupakan salah satu fungsi yang sangat penting untuk kita pelajari. Jika kita dapat mengingat cara menggunakan help(), kita memegang kunci untuk memahami sebagian besar fungsi lainnya. Karena fungsi help() adalah suatu fungsi untuk memahami fungsi semua fungsi.  
  
Berikut ini sebuah contoh:

>>>print(help(len))

Help on built-in function len in module builtins:

len(obj, /)

Return the number of items in a container.

help () menampilkan dua hal:

1. header : header adalah bagian pertama dari output help, dari contoh diatas yaitu bagian len(obj, /). Baris ini memberitahu kita bahwa fungsi len ini menerima argument sebuah object.  
2. Deskripsi singkat tentang apa fungsinya.

## Mendefinisikan fungsi

Python memiliki banyak sekali fungsi bawaan yang sangat membantu kita. Namun kita juga bisa membuat fungsi kita sendiri untuk melangkah lebih jauh dalam membuat suatu aplikasi. Berikut adalah contoh membuat fungsi di python :

def least\_difference(a, b, c):

diff1 = abs(a - b)

diff2 = abs(b - c)

diff3 = abs(a - c)

return min(diff1, diff2, diff3)

Dari contoh di atas, kita membuat fungsi yang bernama least\_difference, dimana fungsi ini memiliki tiga argument, yaitu a, b, c

Fungsi dimulai dengan kata kunci ‘def’. Blok kode yang menjorok setelah tanda ‘:’ dijalankan ketika fungsi dipanggil.

‘return’ adalah kata kunci lain juga yang berkaitan dengan fungsi. Kata kunci ‘return’ adalah untuk menentukan apa yang akan di hasilkan dari fungsi tersebut.

## Default Arguments

Perhatikan fungsi berikut :

def greet(who="Colin"):

print("Hello,", who)

Dari contoh di atas, kita mendefinisikan fungsi yang memiliki arguments ‘who’. Di dalam fungsi tersebut kita mengassign nilai “Colin” terhadap argumen ‘who’. Ini menunjukan, jika kita tidak memberikan nilai ‘who’ ketika kita memanggil fungsi tersebut.,maka fungsi tersebut akan memiliki nilai argument ‘who’ sebagai “Colion”. Berikut contohnya :

# memanggil fungsi tanpa memberikan nilai untuk argumentt who

>>>print(greet())

"Hello Colion"

# memanggil fungsi dengan memberikan argument untuk nilai who

>>>print(greet("Fauzan"))

"Hello Fauzan"

## Method

Method adalah salah satu hal yang sangat penting dalam python. memahami method akan membuat kita semakin pro dalam menggunakan bahasa python.

Sederhananya, method adalah suatu fungsi yang dimiliki oleh suatu object.

Apa itu object? Segala yang ada di python adalah object. Contohnya kita membuat variabel bertipe string, bertipe list, bertipe numeric dan lain sebagainya, itu semua adalah object, dan setiap object memiki fungsi yang hanya bisa digunakan oleh object tersebut. Sebagai contoh, object string memiliki method uppercase, dimana method upper ini tidak bisa digunakan oleh object lain seperti list. Tapi list juga memiliki fungsi seperti index, yang dimana fungsi index ini tidak bisa digunakan oleh data dengan tipe objek lain seperti string contohnya. Jadi intinya, fungsi yang dimiliki oleh suatu object dinamakan method. Berikut adalah contohnya :

# membuat object string

>>> huruf\_kecil = 'huruf\_kecil'

# memanggil salah satu method string

>>> print(huruf\_kecil.upper())

'HURUF\_KECIL'

# mendifinisikan object list

>>>keluarga\_ucup = ['mamah', 'papah', 'ucup', 'adek ucup', 'kaka ucup']

# memanggil method yang dimiliki oleh list

>>>print(keluarga\_ucup.index('ucup'))

2

## jika kita menggunakan method upper ketika object list, maka akan error

## karena method tersebut bukan dimiliki object list

>>>print(keluarga\_ucup.upper())

AttributeError Traceback (most recent call last)

<ipython-input-12-13e588d6418a> in <module>

----> 1 keluarga\_ucup.upper()

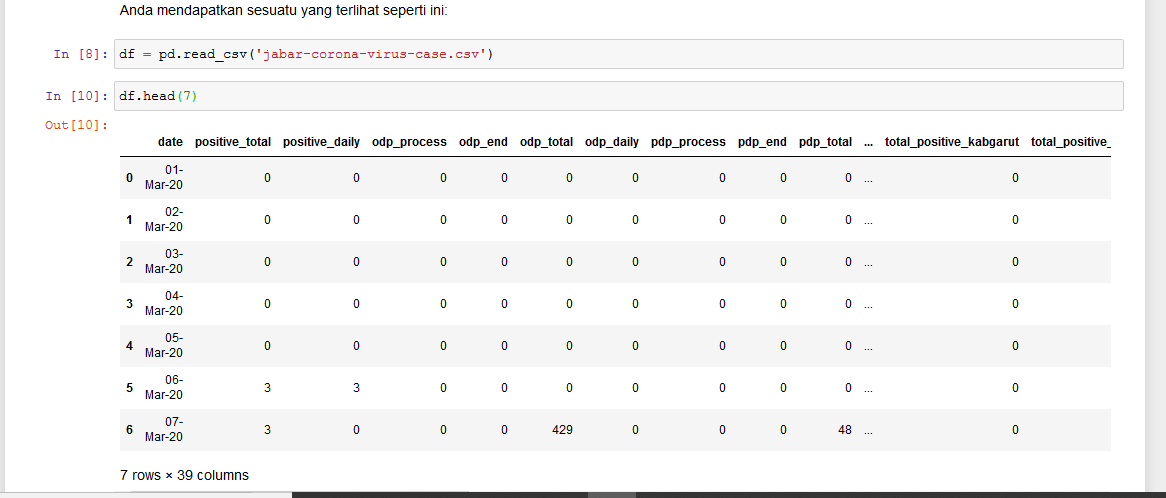
AttributeError: 'list' object has no attribute 'upper'

## Menggunakan External Library

So far, kita telah mengenal fungsi bawaan dari python. Fungsi bawaan adalah fungsi yang sudah ada di python tanpa kita harus menginstall apapun. Namun di python juga ada berbagai fungsi, method, object yang bukan merupakan bawaan dari python, tapi hasil kerja dari orang lain yang telah membagikannya kepada kita, sesuatu ini kita namakan external library. Kita bisa menggunakan external library tersebut dengan cara menginstallnya dan kemudian menggunakannya di kode kita.

Ada banyak sekali external library di python. Beberapa external library yang sangat penting untuk data science adalah numpy, pandas, skit-learn, dan lain sebagainya.

Cara untuk menginstall external library ini, yang paling mudah adalah dengan menggunakan package manager. Di python package manager ini bernama ‘pip’, contohnya bila kita ingin menginstsall library numpy, maka kita cukup membuka command prompt lalu ketikan ‘pip install numpy’, seperti contoh berikut :



Untuk menggunakan external library di kode python kita, kita terlebih dahulu memanggil library tersebut menggunakan suatu keyword yaitu ‘import’. Berikut adalah contoh memanggil dan menggunakan external library di python :

# Definition of radius

>>>r = 0.43

# Import the math package

>>>import math

>>>pi=math.pi

# Calculate C

>>>C = 2\*pi\*r

# Calculate A

>>>A = pi\*r\*r

# Build printout

>>>print("Circumference: " + str(C))

Circumference: 2.701769682087222

>>>print("Area: " + str(A))

Area: 0.5808804816487527

Dari kode block di atas kita memanggil library math, kemudian kita menggunakan attribute pi yang ada di dalam library math.

## External Resource

Untuk lebih memahami lagi materi, silakan kunjungi artikel dan video di bawah ini :

List Artikel :

* <https://belajarpython.com/tutorial/fungsi-python>
* <https://www.petanikode.com/python-fungsi/>

List Videos :

* <https://www.youtube.com/watch?v=WjM68icSw3s&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=12> ( Pengenalan Fungsi )
* <https://www.youtube.com/watch?v=vWuSLG_6rxA&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=13> ( Fungsi Argument )
* <https://www.youtube.com/watch?v=23dDEp6WPH4&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=14> ( Return Value Function )

# Terlindungi: Day 4 – Logic Control Flow and Loop

Python memiliki sebuah tipe data bernama boolean, yang hanya memiliki 2 nilai, yaitu **True**atau **False**.

>>>x = True

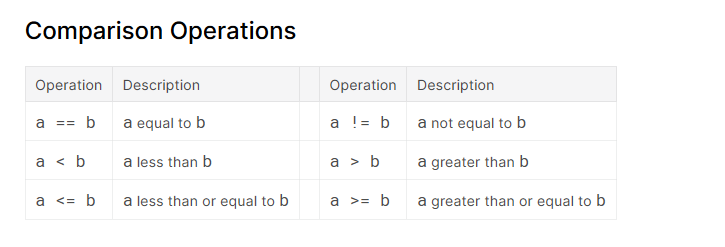
>>>print(x)

>>>print(type(x))

True

<class 'bool'>

Daripada memasukan nilai boolean (True or False) langsung kepada variable, lebih baik mendapatkan nilai dengan melakukan suatu operasi. Operasi itu dinamakan **comparison operator**. Comparison operator ini yang nantinya akan memberikan jawaban nilai berdasarkan logic yang kita bangun. Ada beberapa dasar comparison operator, yaitu sebagai berikut :



def can\_run\_for\_president(age):

"""Can someone of the given age run for president in the US?"""

# The US Constitution says you must "have attained to the Age of thirty-five Years"

return age >= 35

print("Can a 19-year-old run for president?", can\_run\_for\_president(19))

print("Can a 45-year-old run for president?", can\_run\_for\_president(45))

result :

Can a 19-year-old run for president? False

Can a 45-year-old run for president? True

Comparison cukup pintar juga, perhatikan :

print(3.0 == 3)

True

Tapi tidak terlalu pintar

print('3'==3)

False

Python menyediakan operator untuk menggabungkan nilai boolean menggunakan konsep bahasa standar yang biasa kita pahami seperti “and”, “or”, dan “not”.  
  
Dengan ini, kita dapat membuat fungsi seperti ini :

def can\_run\_for\_president(age, is\_natural\_born\_citizen):

"""Can someone of the given age and citizenship status run for president in the US?"""

# The US Constitution says you must be a natural born citizen \*and\* at least 35 years old

return is\_natural\_born\_citizen and (age >= 35)

print(can\_run\_for\_president(19, True))

print(can\_run\_for\_president(55, False))

print(can\_run\_for\_president(55, True))

result :

False

False

True

Boolean dan conditional operator sangat berguna dalam mengatur alur atau logic di dalam kode kita. Namun Boolean dan conditional akan lebih powerful ketika kita menggunakannya bersama **conditional statement**, dengan menggunakan keyword **if, else,**dan **elif**. Conditional statement biasa disebut dengan if-then.

Conditional Statement, memungkinkan programmer untuk mengeksekusi suatu kode tertentu tergantung pada beberapa kondisi Boolean. Contoh dasar dari pernyataan conditional Python adalah ini:

def inspect(x):

if x == 0:

print(x, "is zero")

elif x > 0:

print(x, "is positive")

elif x < 0:

print(x, "is negative")

else:

print(x, "is unlike anything I've ever seen...")

inspect(0)

inspect(-15)

result :

0 is zero

-15 is negative

## Loop

Loop adalah cara untuk berulang kali mengeksekusi beberapa kode. Ini sebuah contoh:

planets = ['Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune']

for planet in planets:

print(planet, end=' ') # print all on same line

result :

Mercury Venus Earth Mars Jupiter Saturn Uranus Neptune

Ketika kita menggunakan loop, ada beberapa hal yang perlu diperhatikan :

1. Kita perlu memberikan nama variable yang akan digunakan (dalam contoh di atas adalah planet)
2. Serentetan nilai yang ingin kita looping untuk melakukan operasi kepada setiap elemennya ( dalam contoh di atas adalah planets ).

**range()**, range() adalah suatu fungsi yang menghasilkan suatu baris nilai. Lebih jauhnya lagi ,kita bisa menggunakan fungsi help yang telah kita pelajari sebelumnya untuk lebih faham berbagai macam cara untuk menggunakannya. Berikut merupakan satu contoh sederhana :

for i in range(5):

print("Doing important work. i =", i)

result :

Doing important work. i = 0

Doing important work. i = 1

Doing important work. i = 2

Doing important work. i = 3

Doing important work. i = 4

Di python ada sebuah fungsi bawaan yang bernama **enumerate()**. Enumerate() memungkinkan kita untuk melakukan loop terhadap suatu object semacam list disertai dengan pengambilan index dari setiap elemennya. Contohnya seperti berikut :

# areas list

areas = [11.25, 18.0, 20.0, 10.75, 9.50]

# Change for loop to use enumerate() and update print()

for index,area in enumerate(areas) :

print("room"+str(index)+": "+str(area))

result :

room0: 11.25

room1: 18.0

room2: 20.0

room3: 10.75

room4: 9.5

Jenis loop lain dalam python adalah **while loop**, yang terus menerus melakukan looping sampai memenuhi suatu kondisi yang membuat dia harus berhenti, berikut contohnya :

i = 0

while i < 10:

print(i, end=' ')

i += 1

result :

0 1 2 3 4 5 6 7 8 9

Argumen dari while loop dievaluasi setiap looping, dan loop dijalankan sampai hasil evaluasi dari conditional operator bernilai False.

## External Resource

List Artikel :

* <https://www.petanikode.com/python-percabangan/>
* <https://www.petanikode.com/python-perulangan/>

List Videos :

* <https://www.youtube.com/watch?v=Hqndpzj0ZFg&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=6> ( Percabangan 1)
* <https://www.youtube.com/watch?v=f28RoIcHZhY&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=7> ( Percabangan 2)
* <https://www.youtube.com/watch?v=KMmZo_dvmyk&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=8> ( Loop 1 )
* <https://www.youtube.com/watch?v=L5GGd1JHqnE&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=9> ( Loop 2 )
* <https://www.youtube.com/watch?v=sLxR7vvPemY&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=10> ( Loop 3 )
* <https://www.youtube.com/watch?v=S8PxQTcme9k&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=11> ( Loop 4)
* <https://www.youtube.com/watch?v=ZnBZWAUusj8&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=22> (Optional : Materi Pengayaan)

# Terlindungi: Day 5 – Python Dictionary

Dictionary adalah suatu topik yang sangat penting dalam python dan data science. Karena dictionary seperti penyusun untuk suatu objek yang lebih kompleks seperti pandas dataframe yang akan kita pelajari nanti.

Jadi, dictionary ini adalah salah satu jenis tipe data di python yang memetakan antara key dan value dari suatu data. Berikut contohnya :

numbers = {'one':1, 'two':2, 'three':3}

Kita dapat mengakses suatu nilai dalam dictionary dengan cara seperti ini :

>>>print(numbers['one'])

1

Kita juga dapat mengubah suatu nilai dalam dictionary dengan cara berikut :

>>>numbers['one'] = 'satu'

>>>print(numbers)

{'one':satu, 'two':2, 'three':3}

Perhatikan value dari key ‘one’ berubah dari awalnya 1 menjadi ‘satu’

Jika kita melakukan loop pada suatu dictionary, maka kita akan me-loop terhadap key pada dictionary tersebut :

or k in numbers:

print("{} = {}".format(k, numbers[k]))

result :

one = satu

two = 2

three = 3

Object dictionary mempunyai suatu method yang bernama items(), dimana dengan fungsi ini kita dapat melakukan loop terhadap suatu dictionary beserta dengan key dan value nya :

or k, v in numbers.items():

print("{} = {}".format(k, v))

result :

one = satu

two = 2

three = 3

Untuk memahami lebih jauh lagi tentang dictionary, kita bisa menggunakan fungsi help sebagai penolong.

## External Resource

Untuk lebih memahami lagi tentang materi ini, kunjungi artikel/video dari link-link di bawah ini :

List Artikel/Tutorial :

* <https://belajarpython.com/tutorial/dictionary-python>
* <https://www.petanikode.com/python-dictionary/>

List Video :

* <https://www.youtube.com/watch?v=ARfcEqYpzkk&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=21> ( Dictionary )

# Terlindungi: Mini-Project 1 : Remove Duplikat List

Buatlah sebuah fungsi, yang menerima suatu argument objek list, dan return sebuah objek list baru, dimana objek list baru ini berisi list sebelumnya dikurangi dengan data yang duplikat, sehingga setiap element di dalam list adalah unik.

* Berikan 2 solusi, solusi pertama menggunakan loop dan list, solusi yang kedua menggunakan objek set!

**New Concept :** Set  
**Resource Materi**tentang Set:

* <https://www.youtube.com/watch?v=Fn6073uEifE&list=PLZS-MHyEIRo7cgStrKAMhgnOT66z2qKz1&index=20>

Example :

# solusi tanpa menggunakan set

def remove\_duplicate(obj\_list):

''''

code Here

''''

return new\_list

# solusi dengan menggunakan set

def remove\_duplicate\_with\_set(obj\_list):

''''

code Here

''''

return new\_list

obj\_list = [1, 2, 4, 6, 2, 1, 4, 5, 7, 8, 6]

print(remove\_duplicate(obj\_list))

print(remove\_duplicate\_with\_set(obj\_list))

output :

[1, 2, 4, 6, 5, 7, 8]

[1, 2, 4, 5, 6, 7, 8]

# Terlindungi: Mini Project 2 : Membuat ChatBot Sederhana

## Persoalan

Buah sebuah fungsi chatbot yang memberikan suatu **respon** dimana chatbot ini menerima **satu argument string**.

Chatbot ini akan **mengecek terlebih dahulu** apakah argument ini adalah **sebuah pertanyaan** atau **pernyataan.**

Jika argument adalah **pertanyaan**kemudian chatbot mengecek lagi apakah pertanyaan tersebut adalah pertanyaan**yang bisa di jawab atau tidak**. **Jika**pertanyaan maka fungsi akan memberikan jawaban yang bersifat **random**berdasarkan **opsi**list jawaban yang disediakan berdasarkan jenis pertanyaan tersebut. **Jika tidak** maka chatbot akan memberikan **jawaban default** bahwa chatbot tidak mengerti pertanyaannya.

Jika argument adalah bukan pertanyaan maka chatbot akan memberikan sebuah statement random yang disediakan.

Untuk memulai mengerjakan soal ini, download file berikut :

## Lingkup Materi :

* String
* Dictionary
* If-elif-else
* Fungsi
* List
* Library/Module Random

## Simulasi Expected Result :

# import library

from datetime import datetime

import random

# ganti dengan sebuah nama

nama = "Nama Anda"

# variabel tanggal

tanggal = datetime.now().day

# default variabel untuk pertanyaan tidak diketahui

default = "maaf, aku tidak tahu jawaban dari pertanyaanmu"

# Membuat objek dictionary berisi berbagai opsi jawaban

# list jawaban untuk pertanyaan tentang nama

jawaban\_nama = [

"nama saya {0}".format(nama),

"orang-orang memanggil saya {0}".format(nama),

"panggil saja saya {0}".format(nama)

]

# list jawaban untuk pertanyaan tentang tanggal

jawaban\_tanggal = [

"hari ini tanggal {0}".format(tanggal),

"ya ampun masa tidak tahu, hari ini tanggal".format(tanggal)

]

# opsi pertanyaan yang bisa dijawab

pertanyaan = {

"nama kamu siapa?": jawaban\_nama,

"kamu siapa?" : jawaban\_nama,

"tanggal berapa hari ini?": jawaban\_tanggal,

"hari ini tanggal berapa?" : jawaban\_tanggal,

"default": default

}

# list jawaban untuk sebuah argument selain pertanyaan

statement = [

'ceritakan lebih banyak!',

'kenapa kamu berpikir begitu?',

'sudah berapa lama kamu merasa seperti ini?',

'Itu sangat menarik!',

'oh wow!',

':)'

]

# respon keseluruhan

responses = {

'pertanyaan' : pertanyaan,

'statement' : statement

}

#------

# ayo buat chatbotmu

def chatbot(message):

## code disini

print(chatbot('Selamat Pagi'))

print(chatbot('Mau bermain bersamaku?'))

print(chatbot('nama kamu siapa?'))

print(chatbot('hari ini tanggal berapa?'))

output :

:)

maaf, aku tidak tahu jawaban dari pertanyaanmu

orang-orang memanggil saya Nama Anda

ya ampun masa tidak tahu, hari ini tanggal

## Penjelasan

* Pada code yang ini print(chatbot(‘Selamat Pagi’)), chatbot melihat list opsi jawaban yang ada pada variabel statement, kemudian memilih secara random jawaban yang akan diberikan.
* Pada code yang ini print(chatbot(‘Mau bermain bersamaku?’)), chatbot melihat bahwa pesan ini adalah sebuah pertanyaan karena di akhiri dengan tanda tanya, kemudian value dari key ‘default’ karena pertanyaan tersebut tidak bisa dijawab oleh chatbot.
* Pada code yang ini print(chatbot(‘nama kamu siapa?’)), chatbot melihat bahwa pesan ini adalah sebuah pertanyaan karena di akhiri dengan tanda tanya, kemudian melihat pada variabel ‘jawaban\_nama’ dan memilih secara random jawaban yang akan di berikan.
* Pada code yang ini print(chatbot(‘hari ini tanggal berapa?’)), chatbot melihat bahwa pesan ini adalah sebuah pertanyaan karena di akhiri dengan tanda tanya, melihat pada variabel ‘jawaban\_tanggal’ dan memilih secara random jawaban yang akan diberikan.