jupyter

Beberapa bagian yang paling penting di Jupyter Notebook yang kita akan pakai di buku ini adalah menu dan cell. Di Menu ada beberapa tombol untuk membantu kita melakukan macem-macem hal seperti save, restart session, convert, insert cell dan lain-lainnya. Jika anda sudah sering pakai Jupyter Notebook lama-lama akan lebih sering pakai *short-cut* daripada menu. Namun untuk awal-awal, ada baiknya sering lihat fungsi-fungsi yang ada di menu.

Di Jupyter Notebook, anda akan mengetik kode-kode yang diperlukan di sebuah kotak yang bernama "cell". Di buku ini, kita hanya memerlukan dua tipe cell yaitu "code" dan "markdown". Anda memerlukan tipe cell "code" untuk mengetik kode yang akan dijalankan oleh python. Jika kode yang anda masukkan dapat dibaca oleh pyton, maka hasilnya akan muncul di output. Output itu bukan sesuatu yang anda ketik, tapi sesuatu yang muncul setelah anda menjalankan (*run*) kode anda di cell code.

Untuk menjalankan sebuah kode, anda ketik kode-nya di cell , lalu klik tombol {{< icon name="play" pack="fas" >}} run, atau dengan *short-cut* ctrl+enter atau shift+enter.

Sementara itu, anda dapat menggunakan cell "markdown" untuk tujuan mengetik. Cell markdown tidak bisa di-jalankan dan tidak akan mengeluarkan output apapun.

Setiap cell sendiri memiliki dua mode, yaitu command mode dan edit mode. ketika cell anda berwarna hijau, artinya anda sedang di mode edit. Anda dapat mengetik di cell yang sedang aktif. Ketika cell berwarna biru, artinya anda sedang di mode command. Mode command membuat *keyboard* anda mengaktifkan *shortcut* dan bukan mengetik.

Untuk masuk ke edit mode, anda cukup me-klik cell yang ingin anda edit. Jika anda klik area lain di luar cell, maka anda akan masuk ke command mode. *Short-cut* untuk masuk ke edit mode adalah enter, sementara untuk ke command mode adalah escape.

## menyimpan jupyter notebook[¶](#menyimpan-jupyter-notebook)

File jupyter notebook akan secara otomatis tersimpan dengan ekstensi .ipnyb. Untuk menyimpannya dalam bentuk pdf, gunakan file -> download as -> .html anda akan memiliki file .html. Buka file tersebut dengan browser anda (chrome, edge, safari, mozila) lalu tekan ctrl+p lalu print to pdf.

Yuk kita mulai menulis kode.

## Kode-kode Sederhana[¶](#Kode-kode-Sederhana)

###### beberapa hal yang dapat anda lakukan[¶](#beberapa-hal-yang-dapat-anda-lakukan)

Silakan coba-cobain nge-run sembarangan. Anda dapat mendelete cell dengan tekan tombol d dua kali di command mode.

In [3]:

# Menulis angka  
1

Out[3]:

1

In [4]:

# fungsi tambah-tambahan  
1+1

Out[4]:

2

In [5]:

# menulis kalimat harus menggunakan tanda petik  
"apa kabar dunia?"

Out[5]:

'apa kabar dunia?'

In [6]:

# tes logika (boolean)  
1 > 2

Out[6]:

False

###### menempelkan informasi ke dalam sebuah obyek[¶](#X87841d7e4ecfcfaa0c540eefdc2534584d77cfe)

Salah satu kelebihan python adalah betapa mudahnya melekatkan informasi (dalam bentuk kode/ekspresi/data) kepada sebuah obyek. Anda melekatkan informasi dengan menggunakan tanda =

In [8]:

# lekatkan '1+1' kepada obyek bernama 'x'  
x=1+1

seperti anda lihat di atas, jika 1+1 kita lekatkan ke sebuah obyek, maka outputnya tidak langsung dicetak. anda harus panggil dulu obyek tersebut dengan mengetik namanya, atau menggunakan fungsi print()

In [9]:

# memunculkan output x  
x

Out[9]:

2

In [16]:

# memunculkan output x dengan fungsi print  
print(x)

2

Dalam hal ini, menggunakan fungsi print() menghasilkan output yang sama persis dengan mengetik langsung nama obyeknya. Tapi nanti kita akan melihat beberapa kegunaan print() yg lain ketika obyek kita isinya lebih ribet daripada sekedar 1+1

Anda juga dapat memanipulasi obyek x dengan operasi matematika sepert + dan - dan lainnya.

In [10]:

# Menambahkan x dengan 3  
x+3

Out[10]:

5

In [11]:

# menempelkan x+3 ke obyek lain dan langsung mencetaknya   
y=x+3  
y

Out[11]:

5

In [14]:

# menambahkan 2 obyek dan melekatkannya ke obyek ke-3  
z = x + y  
z

Out[14]:

6

ketikan setelah tanda #pagar akan diabaikan oleh python. Peneliti biasanya menggunakan tanda #pagar untuk memberi komentar atau keterangan tentang kode yang mereka jalankan. Jika anda menggunakan Jupyter Notebook untuk mengerjakan skripsi atau tugas kuliah, saya sangat merekomendasikan menggunakan tanda #pagar untuk memberi keterangan tentang kode anda kepada teman atau pemeriksa.

Jadi jika anda ingin menulis komentar atau keterangan tentang kode yang anda tulis, gunakan tanda #pagar

In [15]:

z = z-6 # mengubah Z, sebelumnya z=5  
z # y sudah diupdate dengan mengurangkan dengan 6

Out[15]:

0

## Tentang fungsi[¶](#Tentang-fungsi)

kita akan menggunakan banyak sekali fungsi ketika memanipulasi obyek di python, seperti print(). Seiring dengan mengikuti buku ini, kita akan melihat beberapa fungsi lain yang esensial untuk melakukan analisis statistika.

Jika anda tidak mengerti sebuah fungsi tuh sebenernya melakukan apa, anda dapat menggunakan tanda ? setelah mengetik fungsi. Pastikan anda terhubung dengan internet.

Coba anda jalankan kode print? dan lihat apa yang terjadi.

## Tentang paket / modul[¶](#Tentang-paket-/-modul)

Di python ada yang disebut "package" atau "module", saya bahasa Indonesiakan menjadi paket atau modul. Paket-paket ini berisi fungsi-fungsi yang aslinya tidak ada di python, tapi setelah dipanggil / diimpor, jadi bisa kita gunakan. Beberapa paket ini sangat esensial untuk *data science*, seperti pandas dan matplotlib.

kita memanggil paket ini dengan kode import. Contoh, mari kita gunakan paket sys untuk melihat versi python yang kita gunakan di komputer kita

In [26]:

import sys  
sys.version

Out[26]:

'3.7.7 (default, Apr 15 2020, 05:09:04) [MSC v.1916 64 bit (AMD64)]'

fungsi sys.version tidak ada di python original. Kita harus memanggilnya terlebih dahulu dengan menggunakan import sys supaya dapat menggunakan fungsi-fungsi yang ada sys-nya. Import paket tidak perlu dilakukan di setiap cell. Cukup sekali saja untuk setiap sesi. Tapi jika anda buka python baru, anda harus import lagi untuk sesi tersebut.

## Tentang tipe (type) obyek[¶](#Tentang-tipe-(type)-obyek)

Obyek di python dapat digolongkan menjadi beberapa tipe. anda dapat mengetahui tipe sebuah obyek dengan menggunakan fungsi type('obyek'). Contohnya:

In [21]:

type(x)

Out[21]:

int

In [22]:

type([x,y,z])

Out[22]:

list

In [23]:

type("apa kabar dunia?")

Out[23]:

str

In [24]:

type(1<2)

Out[24]:

bool

seperti dapat anda lihat di atas, ada setidaknya 4 tipe obyek yang dikenali oleh python. X, sebuah obyek yang berisi sebuah angka 2, dikenali sebagai int, kependekan dari integer. susunan obyek yang terdiri dari x, y, dan z, dikenali sebagai list. Sementara itu, frasa "apa kabar dunia?" dikenali sebagai str, atau kependekan dari string. Sementara yang terakhir, bool kependekan dari boolean, adalah logika.

Tipe-tipe ini sangat penting untuk diketahui karena kita akan memanipulasi mereka dengan cara yang berbeda-beda.

#### Angka (integer dan float)[¶](#Angka-(integer-dan-float))

Di python, ada dua tipe untuk angka, yaitu integer (int) dan floating (float). Integer adalah bilangan cacah, yaitu bilangan yang terdiri dari angka tanpa koma koma. Float adalah bilangan bulat, yaitu bilangan yang ada koma-koma nya.

Misalnya, Pak Tedjo punya tiga orang anak. artinya, anak pak tedjo adalah obyek bilangan cacah (integer). Di desa Pak Tedjo, kepala keluarganya memiliki rata-rata 2,3 orang anak. Rata-rata anak di desa Pak Tedjo adalah obyek bilangan bulat (float). Tipe yang akan kita gunakan akan tergantung dari jenis data yang kita miliki.

Di Python, kita akan gunakan . sebagai koma. Kenapa? Yah orang barat pakai titik sebagai koma jadi kita ikut aja. Kita gunakan \_ untuk pemisah ribuan.

In [32]:

a=3  
type(a)

Out[32]:

int

In [33]:

b=2.3  
type(b)

Out[33]:

float

interaksi antara integer dan float akan menjadi float

In [35]:

type(a+b)

Out[35]:

float

#### Huruf (Strings)[¶](#Huruf-(Strings))

Di python, obyek yang berisi huruf disebut str atau string. Misalnya data yang isinya nama, maka data tersebut akan dikategorikan sebagai string. Namun demikian, angka juga dapat dikategorikan sebagai string jika angka tersebut tidak memiliki makna hitung apapun. Maksudnya, nomor yang bertipe str tidak dapat dibandingkan maupun ditambah atau dikurangi. Misalnya, orang yang punya nomor rekening 1234 tidak berarti duitnya lebih banyak daripada orang yang nomor rekeningnya 1235. Untuk menyimpan data string, anda harus menggunakan tanda petik. Boleh ', boleh juga ".

In [39]:

type(3)

Out[39]:

int

In [40]:

type('3') # angka 3 akan menjadi string jika dikelilingi dengan tanda petik.

Out[40]:

str

In [41]:

type("3")

Out[41]:

str

In [42]:

3+3

Out[42]:

6

In [43]:

"3" + 3 # angka bertipe string tidak dapat ditambah dengan angka bertipe integer.

---------------------------------------------------------------------------  
TypeError Traceback (most recent call last)  
<ipython-input-43-a62ff2a7686c> in <module>  
----> 1 "3" + 3  
  
TypeError: can only concatenate str (not "int") to str

Anda dapat menambahkan string dengan string di python. Tapi di R tidak bisa yaa

In [46]:

"nama saya adalah" + "Pak Tedjo"

Out[46]:

'nama saya adalahPak Tedjo'

ups. harusnya dikasih spasi. anda juga bisa menambahkan dua obyek string

In [47]:

a = "nama saya adalah " # lihat saya nambahin spasi setelah adalah  
b = "Pak Tedjo"  
a+b

Out[47]:

'nama saya adalah Pak Tedjo'

string dapat dikali dengan integer untuk mengulang string tersebut

In [48]:

a+b\*3 # apa yang terjadi kalau b yg isinya "Pak Tedjo" dikali 3?

Out[48]:

'nama saya adalah Pak TedjoPak TedjoPak Tedjo'

In [49]:

(a+b)\*3 # bagaimana jika (a+b) yang dikali 3?

Out[49]:

'nama saya adalah Pak Tedjonama saya adalah Pak Tedjonama saya adalah Pak Tedjo'

Manipulasi string bisa lebih keren lagi dengan contoh berikut ini:

In [50]:

nama="Pak Tedjo"  
usia=50  
pekerjaan="Pak RT"  
  
tentang\_Tedjo=f"{nama} adalah seorang bapak-bapak berusia {usia} tahun yang bekerja sebagai {pekerjaan}"  
print(tentang\_Tedjo)

Pak Tedjo adalah seorang bapak-bapak berusia 50 tahun yang bekerja sebagai Pak RT

dapatkah anda memahami apa yang terjadi di atas? Apa fungsi dari {}? di tahun depan, usia Pak Tedjo akan bertambah 1 tahun. Bagaimana mengubah "tentang\_Tedjo" yang paling mudah dan cepat?

#### Boolean (logika)[¶](#Boolean-(logika))

Boolean adalah tipe yang bentuknya logika. Biasanya kita menggunakan ini untuk fungsi if, else dan kombinasinya.

In [54]:

t=True  
f=False  
type(t)

Out[54]:

bool

In [55]:

type(f)

Out[55]:

bool

Boolean hanya terdiri dari dua ekspresi, yaitu True dan False. Jangan lupa untuk **tidak** menggunakan tanda petik. Boolean dapat ditambahkan, di mana True bernilai 1 sementara False bernilai 0. Hasilnya akan menjadi int atau float Kita akan lebih jauh mengeksplorasi ini nanti.

In [56]:

t+f

Out[56]:

1

In [57]:

type(t+f)

Out[57]:

int

In [ ]: