

Modul 1

Pendahuluan Pemrograman Python

Setelah mengikuti kuliah ini diharapkan anda bisa menggunakan bahasa pemrograman Python untuk memecahkan masalah di kehidupan sehari-hari (yang bisa dipecahkan dengan pemrograman tentunya).

Persiapan

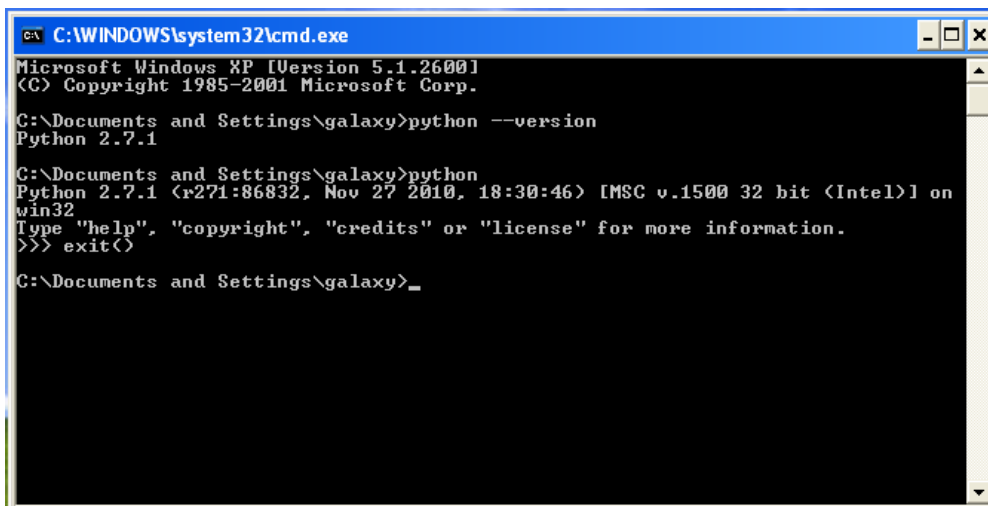
Untuk dapat mengikuti kuliah ini pastikan Python interpreter sudah terinstal di komputer anda. Terdapat dua versi python yang saat ini ada versi 2 vs versi 3. Ada beberapa perbedaan syntax, operasi IO, perubahan struktur modul. Untuk library tambahan Python versi 3 masih kurang daripada versi 3. Jika anda sudah menguasai python 2 akan lebih mudah untuk bermigrasi ke versi 3.

Catatan

Python 2.x is the status quo, Python 3.x is the present and future of the language

Pengguna Windows

Untuk pengguna MS Windows. Python interpreter dapat di download di [Python.org](https://python.org) [Download](#). Kemudian pilih *individual releases*. Ada beberapa alternatif python installer untuk Windows (ActiveState, Enthought). Untuk workshop ini gunakan *default installer* dari python.org.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\galaxy>python --version
Python 2.7.1

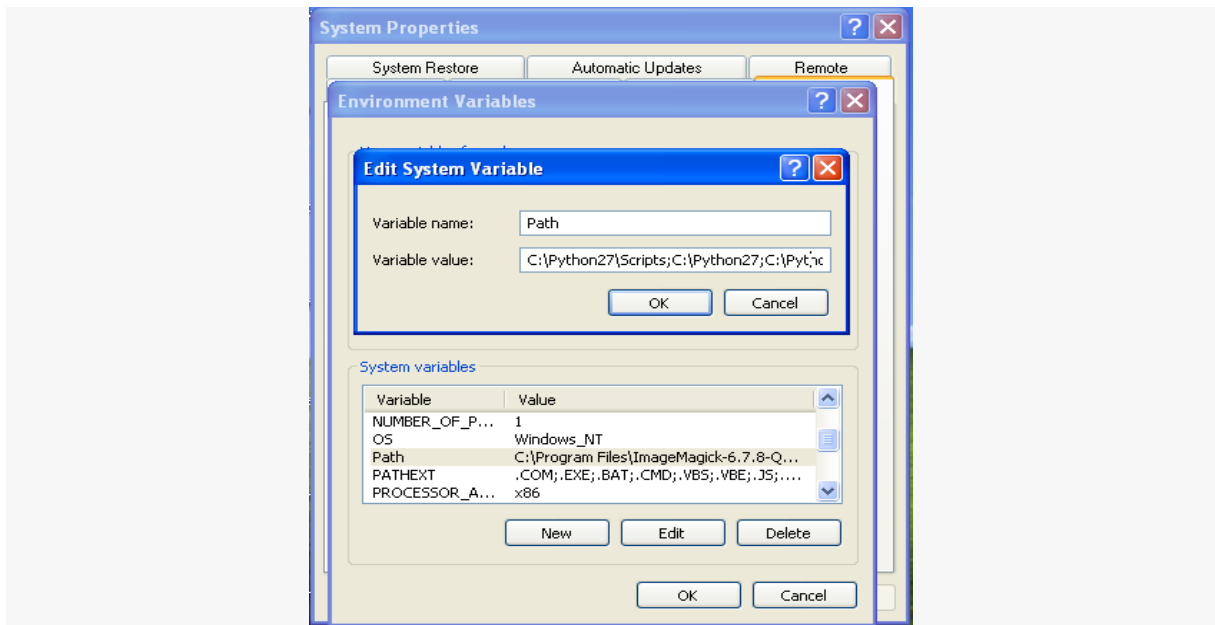
C:\Documents and Settings\galaxy>python
Python 2.7.1 (r271:86832, Nov 27 2010, 18:30:46) [MSC v.1500 32 bit (Intel)] on
win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Documents and Settings\galaxy>_
```

Python di Windows

Catatan

Jika menemukan
kesalahan 'python' is not recognized as an internal or external command, operable program or batch file.
cek windows PATH environment variable



Pengguna Linux

Pada umumnya distro linux sudah menyertakan Python secara default. Pastikan default python interpreter menunjuk ke python versi 2.*:

```
$ python --version
Python 2.7.3
```

```
sakti@neko: ~
~ » python
Python 2.7.3 (default, Aug 1 2012, 05:14:39)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
~ » python --version
Python 2.7.3
~ »
```

Python di Linux

Pengguna Mac OS X

Mac OS X secara default menyertakan Python interpreter.

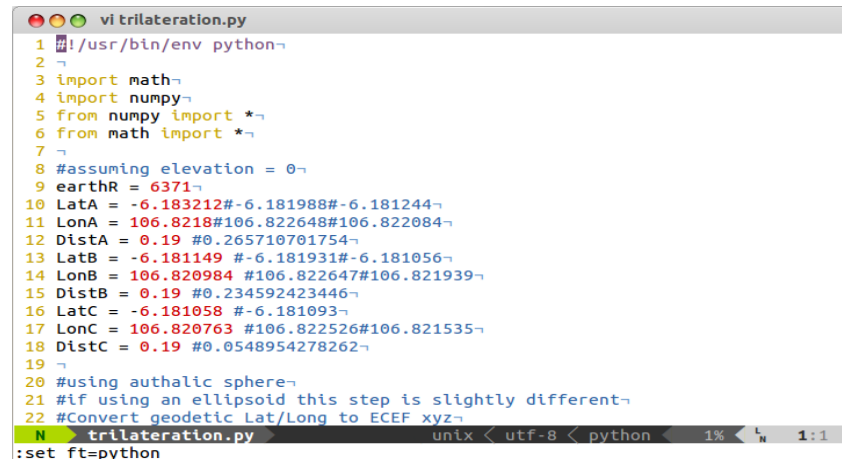
```
sakti-Mac: sakti$ sw_vers
ProductName: Mac OS X
ProductVersion: 10.8.2
BuildVersion: 12S54
sakti-Mac: sakti$ python
Python 2.7.2 (default, Jun 20 2012, 16:23:33)
[GCC 4.2.1 Compatible Apple Clang 4.0 (tags/AppletClang-418.0.60)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
sakti-Mac: sakti$ python --version
Python 2.7.2
sakti-Mac: sakti$ _
```

Python di Mac OS X

Editor Teks

Anda bebas menggunakan editor teks kesayangan anda.

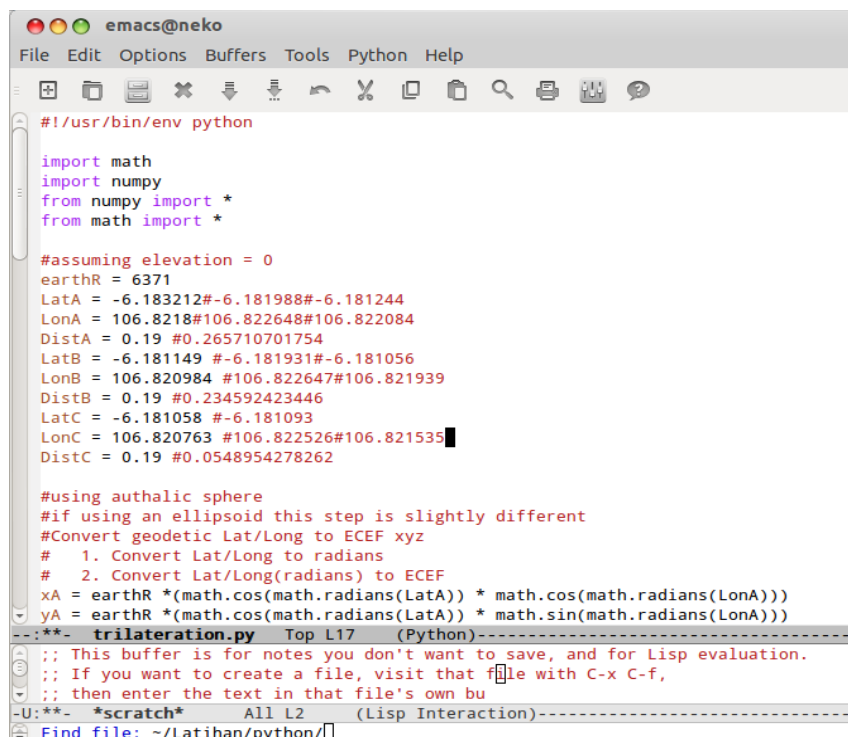
Vim



```
1 #!/usr/bin/env python
2
3 import math
4 import numpy
5 from numpy import *
6 from math import *
7
8 #assuming elevation = 0
9 earthR = 6371
10 LatA = -6.183212#-6.181988#-6.181244
11 LonA = 106.8218#106.822648#106.822084
12 DistA = 0.19 #0.265710701754
13 LatB = -6.181149 #-6.181931#-6.181056
14 LonB = 106.820984 #106.822647#106.821939
15 DistB = 0.19 #0.234592423446
16 LatC = -6.181058 #-6.181093
17 LonC = 106.820763 #106.822526#106.821535
18 DistC = 0.19 #0.0548954278262
19
20 #using authalic sphere
21 #if using an ellipsoid this step is slightly different
22 #Convert geodetic Lat/Long to ECEF xyz
N trilateration.py unix < utf-8 < python < 1% < 1:1
:set ft=python
```

Edit program python menggunakan Vim

Emacs



```
#!/usr/bin/env python

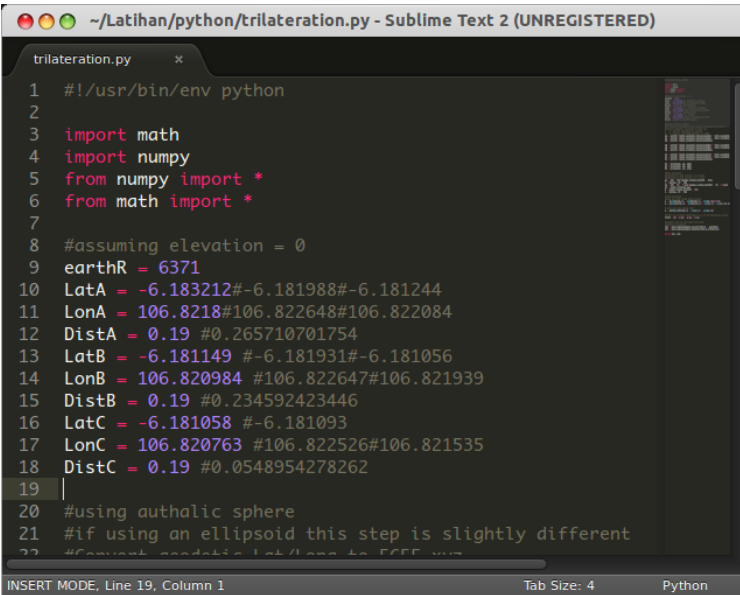
import math
import numpy
from numpy import *
from math import *

#assuming elevation = 0
earthR = 6371
LatA = -6.183212#-6.181988#-6.181244
LonA = 106.8218#106.822648#106.822084
DistA = 0.19 #0.265710701754
LatB = -6.181149 #-6.181931#-6.181056
LonB = 106.820984 #106.822647#106.821939
DistB = 0.19 #0.234592423446
LatC = -6.181058 #-6.181093
LonC = 106.820763 #106.822526#106.821535
DistC = 0.19 #0.0548954278262

#using authalic sphere
#if using an ellipsoid this step is slightly different
#Convert geodetic Lat/Long to ECEF xyz
# 1. Convert Lat/Long to radians
# 2. Convert Lat/Long(radians) to ECEF
xA = earthR *(math.cos(math.radians(LatA)) * math.cos(math.radians(LonA)))
yA = earthR *(math.cos(math.radians(LatA)) * math.sin(math.radians(LonA)))
--:**- trilateration.py Top L17 (Python)-----
;; This buffer is for notes you don't want to save, and for Lisp evaluation.
;; If you want to create a file, visit that file with C-x C-f,
;; then enter the text in that file's own bu
-U:**- *scratch* All L2 (Lisp Interaction)-----
Find file: ~/Latihan/python/
```

Edit program python menggunakan Emacs

SublimeText



```
1  #!/usr/bin/env python
2
3  import math
4  import numpy
5  from numpy import *
6  from math import *
7
8  #assuming elevation = 0
9  earthR = 6371
10 LatA = -6.183212#-6.181988#-6.181244
11 LonA = 106.8218#106.822648#106.822084
12 DistA = 0.19 #0.265710701754
13 LatB = -6.181149 #-6.181931#-6.181056
14 LonB = 106.820984 #106.822647#106.821939
15 DistB = 0.19 #0.234592423446
16 LatC = -6.181058 #-6.181093
17 LonC = 106.820763 #106.822526#106.821535
18 DistC = 0.19 #0.0548954278262
19
20 #using authalic sphere
21 #if using an ellipsoid this step is slightly different
22 #convert geographic lat/long to ECEF xyz
```

INSERT MODE, Line 19, Column 1 Tab Size: 4 Python

Edit program python menggunakan Sublime Text2

Pengantar Python

Dasar-dasar bahasa pemrograman Python.

Menggunakan Python sebagai kalkulator

Program Python dapat dijalankan dengan beberapa mode. Jika kita mengeksekusi Python interpreter tanpa argumen script Python yang telah kita buat, Python interpreter akan masuk ke mode interaktif ([REPL](#), read-eval-print loop).

```
$ python
Python 2.7.3 (default, Aug 1 2012, 05:14:39)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
```

Kita dapat memanfaatkan Python dengan mode interaktif sebagai kalkulator.

```
$ python
Python 2.7.3 (default, Aug 1 2012, 05:14:39)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 1 + 1
2
>>> 40 * 2
80
>>> 40 / 5
8
>>> 9 - 10
-1
>>> 2 + 3 + 4 + 5
14
>>> 2 ** 32
4294967296
```

Operasi aritmatika

Berikut table operasi aritmatika yang ada di Python:

Operasi	Keterangan
+	Menambahkan dua obyek
-	Mengurangi obyek dengan obyek yang lain
*	Perkalian
**	Pangkat
/	Pembagian
//	Pembagian bulat ke bawah
%	Sisa hasil bagi (modulus)

Contoh

Pembagian, perhatikan perbedaan antara bilangan bulat dan pecahan / desimal.

```
>>> 10 / 3
3
>>> 10.0 / 3
3.3333333333333335
>>> 10 / 3.0
3.3333333333333335
>>> 10.0 / 3.0
3.3333333333333335
>>> 10.0 // 3.0
3.0
>>> 10.0 // 3
3.0
```

Sisa hasil bagi.

```
>>> 10 % 3
1
>>> 2 % 3
2
>>> -5 % 4
3
>>> -5 % -4
-1
```

Catatan

mode eksekusi lain:

- m** Mengeksekusi module, contoh: `python -m SimpleHTTPServer` untuk membuat webserver statis
- c** Mengeksekusi command dari parameter yang diterima, contoh: `python -c 'import this'` untuk menampilkan [Zen of Python](#).

Halo Dunia!

Program pertama yaitu program yang jika dijalankan akan mengeluarkan hasil teks berupa `Halo Dunia!`.

```
# lat1.py
print 'Halo Dunia!'
```

Anda bisa membuat file `lat1.py` menggunakan teks editor pilihan anda.

Catatan

Untuk catatan, anda sebaiknya menset teks editor anda agar indentasi menggunakan spasi / space sebanyak 4. Standar PEP (Python Enhancement Proposal) menyarankan agar indentasi selalu konsisten.

Setelah file `lat1.py` disimpan, anda dapat menjalankannya melalui terminal.

```
$ python lat1.py
Halo Dunia!
```

Jika anda menggunakan SublimeText2 anda dapat menjalankannya menggunakan menu `Tools -> Build`, untuk linux anda dapat menggunakan shortcut `Ctrl+b`.



```
/slides/python101/lat/dasar/lat1.py - Sublime Text 2 (UNREGISTERED)
lat1.py
1 # lat1.py
2 print 'Halo Dunia!'

Halo Dunia!
[Finished in 0.0s]

git branch: master, index: 87, working: 4≠ 87, INSERT MODE, Line 1, Column 10 Tab Size
```

Running program python menggunakan SublimeText2

Komentar

Komentar adalah teks apapun yang diawali dengan tanda `#`, digunakan untuk memberikan catatan kepada pembaca kode. Anda dapat melihat kembali `lat1.py`, keterangan keterangan nama file ada dalam bentuk komentar.

Berikut file latihan 2, perhatikan statemen print terakhir tidak akan dieksekusi karena berupa komentar.

```
# lat2.py
# lat2.py adalah nama file ini
# program ini akan menampilkan 'Halo Indonesia!'
# kemudian akan menampilkan 'Halo Jakarta!'

print 'Halo Indonesia!'
print 'Halo Jakarta!'

# print 'Teks ini tidak akan dicetak.'
```

Konstanta Literal

Salah satu contoh konstanta literal yaitu bilangan seperti `5`, `1.23`, atau string seperti `'hari senin'` atau `'hari jum'at'`. Hal ini disebut literal atau harfiah karena anda bisa menggunakan nilai ini secara langsung. Bilangan `2` selalu merepresentasikan dirinya sendiri, dinamakan konstanta karena nilainya tidak dapat diubah.

Dalam latihan 2, `'Halo Indonesia!'` dan `'Halo Jakarta!'` merupakan string literal.

Bilangan

Di Python bilangan dibagi menjadi dua tipe utama - integer (bulat) dan float (pecahan). Salah satu contoh dari integer yaitu 2 yang merupakan bilangan bulat. Contoh untuk float yaitu 3.23 dan 52.3e-4. Notasi e mengindikasikan pangkat 10. Untuk kasus ini 52.3e-4 berarti 52.3×10^{-4} .

String

String adalah rangkaian karakter. Anda bisa menuliskan string literal dengan beberapa cara:

- *Single Quote*

Contoh: 'Halo Bandung!', 'Hari Jum\'at'.

- *Double Quote*

Contoh: "Halo Surabaya!", "Hari Jum\'at". Perhatikan tanda quote ' harus di *escape* pada single quote. Selain itu tidak ada perbedaan antara single quote dan double quote, anda bebas untuk memilih.

- *Triple Quote*

Python mendukung multi-line string atau string dengan baris lebih dari satu. Anda dapat dengan bebas menuliskan single quote ' dan double quote " dalam string literal yang diapit dengan triple quote. Contoh:

```
"""Ini adalah contoh multi-line string
saya tambahkan single quote ' dan double
quote ", tanpa perlu meng-escape || terlebih dahulu"""
```

Contoh lain:

```
'''Ini adalah contoh multi-line string
saya tambahkan single quote ' dan double
quote ", tanpa perlu meng-escape || terlebih dahulu'''
```

Perhatikan perbedaan antara dua contoh diatas.

Immutable

String bersifat immutable yang berarti setelah string dibuat, string tersebut tidak bisa diubah.

Format String

Terkadang kita ingin membuat string dari informasi lain, untuk hal ini kita dapat menggunakan format string.

```
# lat3.py
# format string menggunakan operator '%' dan method format

print '%s pergi ke %s' % ('ibu', 'pasar')
print '{0} pergi ke {1}'.format('ibu', 'pasar')
```



```
print 'jumlah total: %10.3f' % 10.3333
print 'jumlah total: {0:10.3f}'.format(10.3333)
```

Catatan

Operator % jika digunakan untuk string bukan berarti modulus melainkan string format.

Variabel

Hanya menggunakan konstanta literal saja cukup membosankan, kita membutuhkan cara untuk menyimpan dan memanipulasi informasi. Untuk hal ini kita bisa menggunakan variabel. Seperti namanya, variabel dapat diisi dengan bermacam-macam nilai, anda dapat menyimpan apapun menggunakan variabel. Variabel adalah sebagian dari memori komputer yang digunakan untuk menyimpan informasi. Berbeda dengan konstanta literal, anda membutuhkan cara untuk mengakses variabel ini, oleh karena itu kita memberi nama kepada variabel.

Nama Pengenal

Berikut aturan penamaan variabel dalam python.

- Karakter pertama harus berupa karakter alfabet (huruf besar atau huruf kecil ASCII, atau unicode) atau underscore `_`.
- Karakter selanjutnya dapat berupa alfabet (huruf besar atau huruf kecil ASCII, atau unicode), underscore `_` atau digit (0-9).
- Nama variabel bersifat case-sensitif. Sebagai contoh, `namaMhs` dan `namamhs` adalah variabel yang berbeda.

```
# lat4.py
# menggunakan variabel

a = 10
b = 20
c = 30

total = a + b + c

nama = 'ibu'
tempat = 'kantor'

print 'jumlah total = %s' % total
print '%s pergi ke %s' % (nama, tempat)
```

Tipe Data

Variabel dapat menyimpan nilai dengan berbagi tipe disebut dengan tipe data. Bilangan dan string adalah tipe dasar, yang sudah dibahas sebelumnya. Pada latihan berikutnya akan dibahas tipe data yang lain.

Anda menggunakan `type` untuk menentukan tipe data variabel / obyek yang ada.

```
>>> type(1)
<type 'int'>
>>> type(3.2)
<type 'float'>
>>> type(2 ** 1000)
<type 'long'>
>>> type('abc')
<type 'str'>
>>> type('a')
<type 'str'>
```

Obyek

Semua yang ada dalam Python adalah obyek / object. Obyek memiliki field yang memiliki nilai tertentu dan method untuk operasi tertentu.

Untuk melihat field dan method yang ada dalam suatu obyek kita dapat gunakan fungsi builtin `dir`.

```
>>> dir('abc')
['_add__', '__class__', '__contains__', '__delattr__', '__doc__', '__eq__',
'__format__', '__ge__', '__getattr__', '__getitem__', '__getnewargs__',
'__getslice__', '__gt__', '__hash__', '__init__', '__le__', '__len__',
'__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__',
'__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__',
'__sizeof__', '__str__', '__subclasshook__', '_formatter_field_name_split',
'_formatter_parser', 'capitalize', 'center', 'count', 'decode', 'encode',
'endswith', 'expandtabs', 'find', 'format', 'index', 'isalnum', 'isalpha',
'isdigit', 'islower', 'isspace', 'istitle', 'isupper', 'join', 'ljust',
'lower', 'lstrip', 'partition', 'replace', 'rfind', 'rindex', 'rjust',
'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith',
'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
>>> 'abc'.upper
<built-in method upper of str object at 0x7fe601a1f800>
>>> 'abc'.upper()
'ABC'
```

Selain dapat melihat isi obyek, anda dapat mengakses dokumentasi object menggunakan `help`.

```
>>> help(str)
Help on class str in module __builtin__:

class str(basestring)
|   str(object) -> string
|
|   Return a nice string representation of the object.
|   If the argument is a string, the return value is the same object.
|
|   Method resolution order:
|       str
|       basestring
|       object
...

```

```
>>> help(str.upper)
Help on method_descriptor:

upper(...)
    S.upper() -> string

    Return a copy of the string S converted to uppercase.
```

Penulisan Program Python

Berikut cara menulis program Python.

- Buka teks editor pilihan anda, seperti: vim, emacs, gedit, notepad++, sublimetext2.
- Ketikkan kode program seperti contoh yang ada (hindari copy-paste).
- Simpan sesuai nama yang ada.
- Untuk menjalankan program gunakan terminal / command line, ketik `python namaprogram.py`.

Catatan

Untuk pengguna sublimetext2 anda dapat menjalankan program python menggunakan shortcut `Ctrl+b`.

Baris Logis dan Fisik

Baris fisik adalah apa yang anda lihat ketika anda melihat program. Baris logis adalah apa yang Python lihat sebagai statemen tunggal. Python mengasumsikan bahwa setiap baris fisik sesuai dengan baris logic.

Sebagai contoh baris logis seperti statemen `print 'Halo Dunia!'`, jika anda menulis sebagai satu baris maka baris logis sesuai dengan baris fisik.

Catatan

Anda dapat menulis `print 'Halo Dunia!'` menjadi dua baris, contoh:

```
print \
'Halo Dunia!'
```

Anda juga dapat membuat beberapa baris logis menjadi satu baris fisik, contoh:

```
nama = 'budi'; print nama
```

Secara implisit, Python menyarankan menggunakan satu statemen tiap baris untuk menjadikan kode menjadi lebih mudah dibaca.

Indentasi

Karakter spasi penting untuk bahasa pemrograman Python. Lebih tepatnya **spasi diawal baris** atau indentasi. Spasi diawal (baik berupa spasi atau tab) baris logis digunakan untuk menentukan level indentasi, yang akan mempengaruhi pengelompokan statemen.

Statemen yang mempunyai level indentasi sama masuk dalam satu kelompok yang disebut blok / **block**. Hal ini akan digunakan pada bab berikutnya.

```
# lat5.py
```

```
# error indentasi

a = 10
b = 20
c = 30

    total = a + b + c

nama = 'ibu'
tempat = 'kantor'

print 'jumlah total = %s' % total
print '%s pergi ke %s' % (nama, tempat)
```

Latihan:

1. Tuliskan perintah-perintah berikut secara interaktif dan amati hasilnya.

```
>>> 2 + 2
>>> 50 - 5*6
>>> (50 - 5*6) / 4
>>> 8 / 5 # division always returns a floating point number
>>> 17 / 3 # classic division returns a float
>>> 17 // 3 # floor division discards the fractional part
>>> 17 % 3 # the % operator returns the remainder of the
division
>>> 5 * 3 + 2 # floored quotient * divisor + remainder
>>> 5 ** 2 # 5 squared
>>> 2 ** 7 # 2 to the power of 7

>>> width = 20
>>> height = 5 * 9
>>> width * height

STRING :
>>> 'spam eggs' # single quotes
>>> 'doesn\'t' # use \' to escape the single quote...
>>> "doesn't" # ...or use double quotes instead
>>> '"Yes," they said.'
>>> "\"Yes,\" they said."
>>> '"Isn\'t," they said.'

>>> '"Isn\'t," they said.'
>>> print('"Isn\'t," they said.')
>>> s = 'First line.\nSecond line.' # \n means newline
>>> s # without print(), \n is included in the output
>>> print(s) # with print(), \n produces a new line
```

2. Tuliskan kode program berikut dengan text editor dan simpan dengan nama : var.py

```
# Filename : var.py
i = 5
```

```
print(i)
i = i + 1
print(i)

s = '''This is a multi-line string.
This is the second line.'''
print(s)
```

jalankan dengan perintah dengan command window:
python var.py

3. Tuliskan kode program berikut untuk melakukan penjumlahan dua buah bilangan, simpan dengan nama : jumlah.py

```
# filename = jumlah.py
# Memasukkan Inputan Angka

angka1 = input('Tulis angka pertama: ')
angka2 = input('Tulis angka kedua: ')

# Mengkonversi Angka lalu Menjumlahkannya
sum = int(angka1) + int(angka2)

# Menampilkan Hasil Penjumlahan
print('Hasil Penjumlahan {0} dan {1} adalah {2}'.format(angka1,
angka2, sum))
```