Pemrograman dengan Python untuk Pemula

oon arfiandwi

http://oo.or.id/newsletter

Material ini digunakan untuk kelas teknologi pengenalan pemrograman dengan bahasa pengantar Python http://oo.or.id/py

Tools to Install

Download Python 2.7 from http://python.org/download or you can easly use online Python interpreter

For Google Cloud Platform topic:

Google App Engine SDK for Python from https://cloud.google.com/appengine/downloads

Pip Python Package Manager from http://pip.pypa.io/en/latest/installing.html

Outline

Python intro

Type, Variable, Value

Functions

Conditional

Iteration

String processing

Intro to Google Cloud Platform with Python

Reference

http://www.greenteapress.com/thinkpython/

https://developers.google.com/edu/python

http://www.pythonlearn.com/

http://www.diveintopython.net/

https://www.python.org/download/

http://pythoncentral.org/comparison-of-python-ides-development/

Software/Program/App

compiled or interpreted (or both)

input-process-output (decide input & output, think about the process)

syntax and semantic

```
>>> 1 + 2 = 3
>>> panjang = 3
>>> lebar = 4
>>> luas = panjang + lebar
```

Python

- High-level language
- Interactive mode & Scripting mode
- Online interpreter
  - http://www.skulpt.org/
  - http://pythontutor.com/
- case sensitive

Python Brochure | http://getpython.info

Google

«Python has been an important part of Google since the beginning and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we're looking for more people with skills in this language.»

Peter Norvig – *Director of Research at Google Inc*

Powered by Python

In 1998, a small company from California, with the help of the Python programming language, changed the way we search for relevant information on the Internet. The company had an unusual-sounding name: Google.

Blender

It's not just commercial software products that benefit from Python code. Python is also used in many open-source programs such as Blender to automate complex work processes efficiently.

Blender is a free open-source 3D content creation suite, available for all major operating systems under the GNU General Public License.

Python impresses Mozilla

Tarek Ziadé – *member of the Mozilla Service Team, on the reasons for success.*

The Mozilla Corporation, manufacturer of the Firefox web browser and Thunderbird e-mail client, is impressed with Python and uses the language for its web services. The addon.mozilla.org and support.mozilla.com websites and the Socorro crash-reporting system are based on Django and Python.

Firefox Sync Server

The Firefox Sync Server, which is used to synchronize bookmarks, browsing histories, passwords, and open tabs on different computers and mobile devices, was also written in Python.

«The Python programming language supports many programming paradigms and can be put to productive use virtually anywhere. What's more, Python is not restricted to the web. For example, we also use Python for our packaging and build systems.

Introduction to Python with Jessica McKellar

Hello Python

>>> print ‘Hello, World!’

Hello, World!

>>> type(‘Hello, World!’)

<type ‘str’>

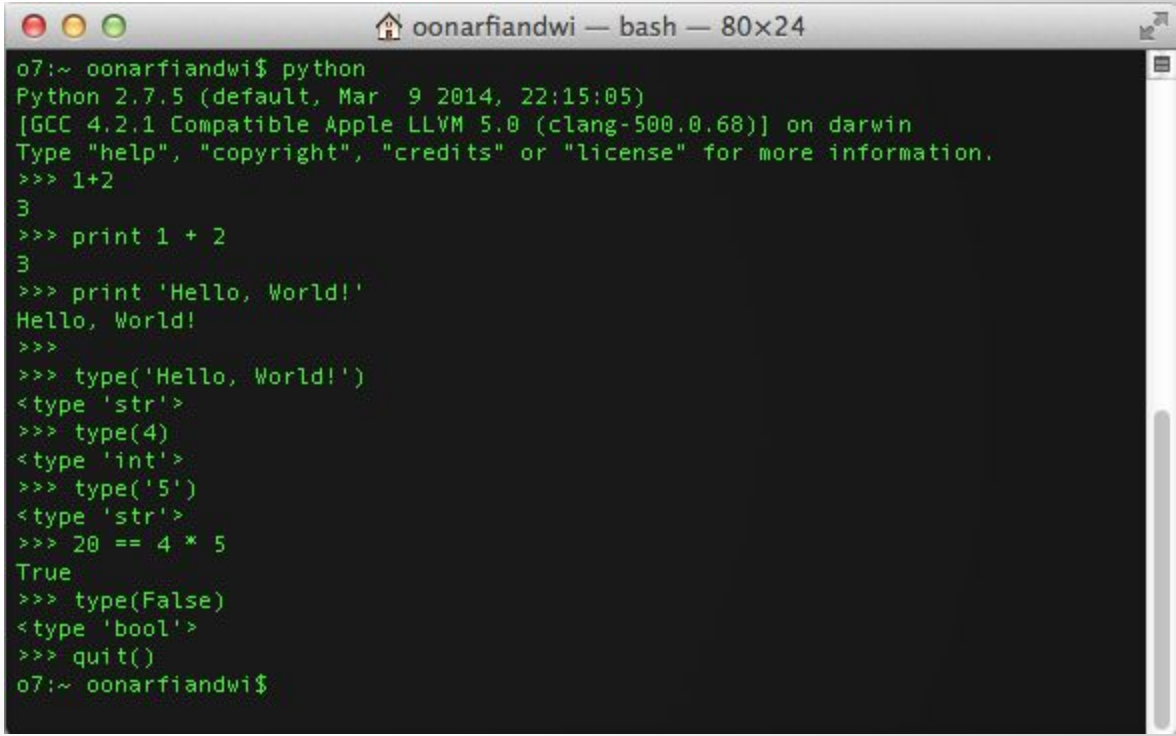
>>> print 1+2

3

>>> type(3)

<type ‘int’>

>>> type(‘1’)



Type, Variable, Value

```
>>> i = 1
```

(i adalah variable, dengan type int, memiliki value 1, diberi nilai dengan operator =)

```
>>> kata = ‘nilai’
```

(kata adalah variable, dengan type str, memiliki value nilai)

some of basic data type:

- Integer: int
- String: str
- Boolean: bool

Variable Assignment

```
>>> message=’message in the bottle’
>>> print message
>>> type(message)
>>> x=1
>>> x=x+2
>>> x+=3
>>> y=4
>>> print x*y
```

Operator & Operand

1+2

(1 adalah operand, + adalah operator, 2 adalah operand)

x < xx and z > y

(operand? operator?)

Order of operations: PEMDAS

Indentation

4-space indents and no hard tab character

Interactive Mode & Scripting Mode

give sample of interactive mode on console

give sample of scripting mode count_down.py

List

List is a sequence, of elements or items

[1, 2, 3]

['ab', 'bc', 'cd', 'de']

a = ['ab', 'bc', 1, 2, 1.1, [True, False]]

print a[5]

Array

array.array is a wrapper of C arrays. can handle only homogeneous C array of data.

so for common use, simply use List

http://stackoverflow.com/questions/176011/python-list-vs-array-when-to-use

Tuple

```
>>> t = (1, 2, 3)
>>> type(t)
<type ‘tuple’>
>>> print t
>>> t = (‘hello’)
>>> print t
```

Set

class set

s = { ‘satu’, ‘dua’, ‘tiga’ }

```
if x in s:
    print ‘found’
```

functions/methods:

len(), union(), intersection(), difference()

Dictionary

class dict

{ key: value }

a = { 1: ‘satu’, 2: ‘dua’, 3: ‘tiga’ }

print a[2]

b = { ‘a’: ’alpha’, ‘b’: ‘beta’, ‘c’: ‘charlie’}

print b[‘b’]

String

String as sequence

String slice

String methods (intro OO first)

String comparison

in Operator

String

```
>>> kata = ‘pemrograman’
>>> print kata[4]
>>> lenkata = len(kata)
>>> print lenkata
>>> print kata[lenkata-1]
>>> for char in kata:
…         print char
>>> print kata.upper()
```

String Operations

Concatenation

>>> print ‘pemrograman’ + ‘python’

Formatted Printing

```
print 'formatting %s %d' % ('text', 7)
```

http://forums.udacity.com/questions/2019688/python-101-unit-2-formatted-printing-and-function-documentation

Conditional

if

if .. else ..

if .. elif .. else ..

Conditional sample

```
>>> hari = ‘minggu’
>>> if hari == ‘sabtu’ or hari == ‘minggu’:
…         print ‘akhir pekan!’
…
akhir pekan!
```

boolean, and, or

x and y

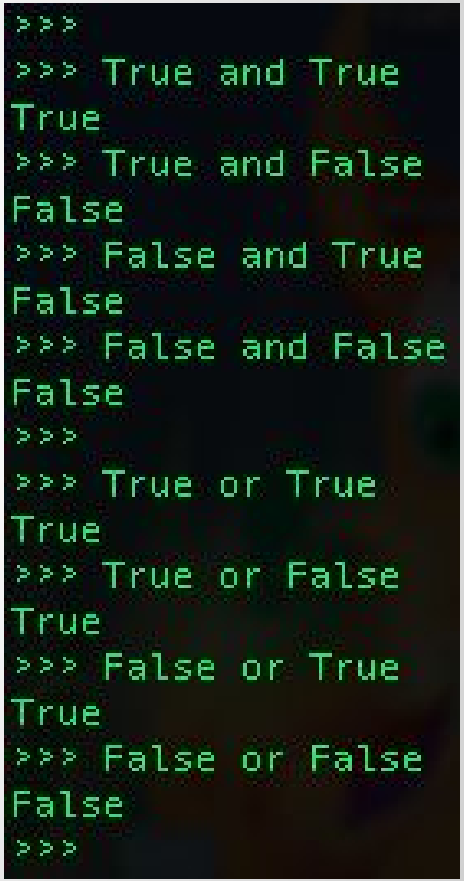
akan bernilai True, jika keduanya True

selain itu bernilai False

x or y

akan bernilai False, jika keduanya False

selain itu bernilai True



Conditional samples (2)

```
>>> x = 4
>>> if x%2 == 0:
...         print ‘even’
...     else:
...         print ‘odd’
```

assignments

buat kode program untuk mengecek apakah variable x itu positif atau negatif

```
>>> x = -1
```

buat kode program untuk mengecek apakah variable x itu positif, negatif, atau nol

```
>>> x = 0
```

Conditional samples (3)

```
>>> x = 7
>>> if x > 8:
...         print ‘A’
...     elif x>6 and x<8:
...         print ‘B’
...     else:
...         print ‘C’
```

Functions

Function call

Add new functions

Fruitful function, void function

import

importing with from

function call and create function

```
>>> type(True)
>>> def apakah_genap(angka):
…         if angka%2 == 0:
…             print ‘benar, genap’
…         else:
…             print ‘salah, ganjil’
>>> apakah_genap(7)
>>> apakah_genap(100)
```

function call and create function

```
>>> def luas_persegi(panjang, lebar):
...         luas = panjang * lebar
...         return luas
...
>>> luasnya = luas_persegi(20, 30)
>>> print luasnya
```

import function

```
>>> import math
>>> print math
>>> akar9 = math.sqrt(9)
>>> print akar9
```

assignment

buat sebuah fungsi luas_lingkaran

>>> luas_lingkaran(10)

note:

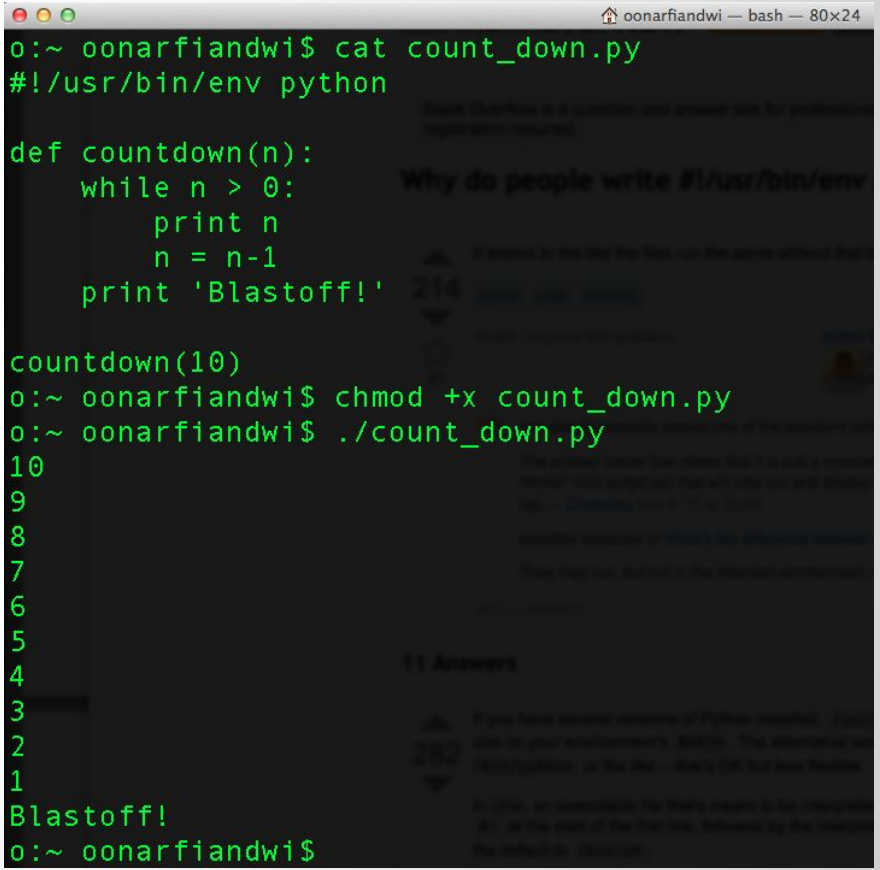
pi: math.pi

input: jari-jari

process: pi * jari-jari * jari-jari

Iteration with while

```
>>> x = 7
>>> while x>0:
...         print ‘data-%i’ % x
...         x = x -1
```



assignment

buat iteration menggunakan while, dalam sebuah fungsi, yang akan menampilkan hanya angka yang genap saja.

```
>>> count_down_genap(4)
4
2
```

Iteration while with break

```
>>> while not ketemu:
...         databerikut = cari_data()
...         if databerikut == yangdicari:
...             break
...
```

Iteration with for

```
>>> for i in range(3)
...         print ‘ulang’
>>> for i in range(5,10)
...         print i
```

comments & docstring

```
>>> xxx = 3  # var name xxx
>>> print xxx

‘’’
   docstring
‘’’
```

Variable Scope

local

global

Google Cloud Platform with Python

let’s start by making a website using python

Google Cloud Platform have a free tier of Google App Engine

we’ll develop using Google App Engine SDK for Python

https://developers.google.com/appengine/docs/python/gettingstartedpython27/introduction

Google App Engine SDK for Python

create 2 files on directory

app.yaml

main.py

dev_appserver.py projectdir

open http://localhost:8080/

app.yaml

https://cloud.google.com/appengine/docs/python/config/appconfig

```
application: waktusaatini
version: 1
runtime: python27
api_version: 1
threadsafe: true

handlers:
- url: .*
  script: main.application

libraries:
- name: webapp2
  version: latest
```

main.py

```
import datetime
import webapp2

class MainPage(webapp2.RequestHandler):
    def get(self):
        message = ‘<p>waktu %s</p>’ % datetime.datetime.now()
        self.response.out.write(message);

application = webapp2.WSGIApplication([(‘/’, MainPage)], debug=True)
```

Template Engine Jinja2

install Jinja2

pip install jinja2

3 Files:

- app.yaml
- main.py
- home.html

Upload to appspot:

appcfg.py update kelas-teknologi/

Sample: http://kelas-teknologi.appspot.com

```
o:kelas-teknologi oonarfiandwi$ ls -la
total 24
drwxr-xr-x  5 oonarfiandwi  staff  170 Nov 29 23:26 .
drwxr-xr-x  6 oonarfiandwi  staff  204 Nov 29 23:25 ..
-rw-r--r--@ 1 oonarfiandwi  staff  253 Nov 29 23:25 app.yaml
-rw-r--r--@ 1 oonarfiandwi  staff  348 Nov 29 23:25 home.html
-rw-r--r--@ 1 oonarfiandwi  staff  812 Nov 29 23:25 main.py
o:kelas-teknologi oonarfiandwi$ cd ..
o:py oonarfiandwi$ ls
clock           kelas-teknologi waktusaatini
o:py oonarfiandwi$ appcfg.py update kelas-teknologi/
11:26 PM Application: kelas-teknologi; version: 1
11:26 PM Host: appengine.google.com
11:26 PM
Starting update of app: kelas-teknologi, version: 1
11:26 PM Getting current resource limits.
Email: kelasteknologi@gmail.com
Password for kelasteknologi@gmail.com:
11:28 PM Scanning files on local disk.
11:28 PM Cloning 3 application files.
11:28 PM Compilation starting.
11:28 PM Compilation completed.
11:28 PM Starting deployment.
11:28 PM Checking if deployment succeeded.
11:28 PM Deployment successful.
11:28 PM Checking if updated app version is serving.
11:28 PM Completed update of app: kelas-teknologi, version: 1
o:py oonarfiandwi$
```

app.yaml

```
application: kelas-teknologi
version: 1
runtime: python27
api_version: 1
threadsafe: true

handlers:
- url: .*
  script: main.application

libraries:
- name: webapp2
  version: latest
- name: jinja2
  version: latest
- name: markupsafe
  version: latest
```

main.py

```
import datetime
import webapp2
import jinja2
import os

from google.appengine.api import users

template_env = jinja2.Environment(
    loader=jinja2.FileSystemLoader(os.getcwd()))
```

main.py (cont’d)

```
class MainPage(webapp2.RequestHandler):
    def get(self):
        current_time = datetime.datetime.now()
        user = users.get_current_user()
        login_url = users.create_login_url(self.request.path)
        logout_url = users.create_logout_url(self.request.path)
        template = template_env.get_template('home.html')
        context = {
            'current_time': current_time,
            'user': user,
            'login_url': login_url,
            'logout_url': logout_url,
        }
        self.response.out.write(template.render(context))

application = webapp2.WSGIApplication([('/', MainPage)], debug=True)
```

home.html

```
<html><head><title>The Time Is...</title></head>
<body>
  {% if user %}
  <p>
    Welcome, <strong>{{ user.email() }}</strong>!
    You can <a href="{{ logout_url }}">sign out</a>.
  </p>
  {% else %}
  <p> Welcome!
    <a href="{{ login_url }}">Sign in or register</a> to customize. </p>
  {% endif %}
  <p>The time is: {{ current_time }}</p>
</body></html>
```

Web Form & Datastore

5 Files:

- app.yaml
- models.py
- main.py
- prefs.py
- home.html

(TODO: models.py should migrate from **db to ndb**)

Upload to appspot:

appcfg.py update kelas-teknologi/

Sample: http://kelas-teknologi.appspot.com

app.yaml

```
application: kelas-teknologi
version: 1
runtime: python27
api_version: 1
threadsafe: true

handlers:
- url: /prefs
  script: prefs.application
  login: required

- url: .*
  script: main.application
```

```
libraries:
- name: webapp2
  version: latest
- name: jinja2
  version: latest
- name: markupsafe
  version: latest
```

models.py

```
from google.appengine.api import users
from google.appengine.ext import db

class UserPrefs(db.Model):
    tz_offset = db.IntegerProperty(default=0)
    user = db.UserProperty(auto_current_user_add=True)

def get_userprefs(user_id=None):
    if not user_id:
        user = users.get_current_user()
        if not user:
            return None
        user_id = user.user_id()

    key = db.Key.from_path('UserPrefs', user_id)
    userprefs = db.get(key)
    if not userprefs:
        userprefs = UserPrefs(key_name=user_id)
    return userprefs
```

main.py

```
...snip...
import models
...snip...

class MainPage(webapp2.RequestHandler):
    def get(self):
        current_time = datetime.datetime.now()
        user = users.get_current_user()
        userprefs = models.get_userprefs()
        if userprefs:
            current_time += datetime.timedelta(0, 0, 0, 0, 0, userprefs.tz_offset)
        ...snip...
        context = {
            'current_time': current_time,
            'user': user,
            'login_url': login_url,
            'logout_url': logout_url,
            'userprefs': userprefs,
        }
        self.response.out.write(template.render(context))

application = webapp2.WSGIApplication([('/', MainPage)], debug=True)
```

home.html

```
<html><head><title>The Time Is...</title></head>
<body>
 {% if user %}
 <p>
  Welcome, <strong>{{ user.email() }}</strong>!
  You can <a href="{{ logout_url }}">sign out</a>.
 </p>
 {% else %}
 <p> Welcome!
  <a href="{{ login_url }}">Sign in or register</a> to customize. </p>
 {% endif %}
 {% if user %}
  <form action="/prefs" method="post">
   <label for="tz_offset">
    Timezone offset from UTC (can be negative):
   </label>
   <input name="tz_offset" id="tz_offset" type="text"
    size="4" value="{{ userprefs.tz_offset }}" /> <input type="submit" value="Set" />
  </form>
 {% endif %}
 <p>The time is: {{ current_time }}</p>
</body></html>
```

prefs.py

```
import webapp2
import models

class PrefsPage(webapp2.RequestHandler):
    def post(self):
        userprefs = models.get_userprefs()
        try:
            tz_offset = int(self.request.get('tz_offset'))
            userprefs.tz_offset = tz_offset
            userprefs.put()
        except ValueError:
            # User entered a value that wasn't an integer. Ignore for now.
            pass

        self.redirect('/')

application = webapp2.WSGIApplication([('/prefs', PrefsPage)],
debug=True)
```

Open Source Project Di.Lingkari.com

web http://di.lingkari.com source code https://github.com/oonid/dilingkari

- Python on Google App Engine
- Flask
- Jinja2
- Google Datastore
- Google+ API
- cron & taskqueue (taskqueue automatic & manual)

Sample with Flask

https://github.com/GoogleCloudPlatform/appengine-python-flask-skeleton

```
# cd appengine-python-flask-skeleton
# pip install -r requirements.txt -t lib
```

main.py (with Flask)

```
# Import the Flask Framework
from flask import Flask
app = Flask(__name__)
# Note: We don't need to call run() since our application is embedded within the App Engine WSGI application server.

@app.route('/')
def hello():
    """Return a friendly HTTP greeting."""
    return 'Hello World!'

@app.errorhandler(404)
def page_not_found(e):
    """Return a custom 404 error."""
    return 'Sorry, Nothing at this URL.', 404

@app.errorhandler(500)
def page_not_found(e):
    """Return a custom 500 error."""
    return 'Sorry, unexpected error: {}'.format(e), 500
```

App Engine Python 2.7 Runtime Lib

django

endpoints

jinja2

webapp2

yaml

MySQLdb

...

https://cloud.google.com/appengine/docs/python/tools/libraries27

Object-Oriented Python

class

inheritance

superclass object

__init__()

method

Next Project: Advanced Class

Machine Learning using Python.

http://tensorflow.org

Projects

Django web based application

    http://djangoproject.com

Kivy cross-platform desktop-android application

    http://kivy.org

Oon Arfiandwi

Co-founder of 7Langit mobile agency

http://oo.or.id/py

http://google.com/+oonarfiandwi

http://twitter.com/oonid

http://about.me/oon

Reference

Green Tea Press, Think Python. June, 2014.

O’Reilly, Programming Google App Engine. October, 2012.

PacktPub, Mastering Object-Oriented Python. April, 2014.

O’Reilly, Flask Web Development. May, 2014.