Pre-class assignments

the “random: pre-class assignments

Write a program in JavaScript (or ANY language)t that performs the following:

1. generate 15 random integers between 1 and 100
2. sort the integers from small to large

SUBMIT on SFS

Programming with Script Languages :(using Python)

Spring 2016
Tuesday 2nd period
Ucu Maksudi

Overview

- Hands-on oriented Python 2 programming
- For developing general applications
- Please check the prerequisite courses:

OOP + Algorithm and Data Structure???

- The first half of the class focuses on the Python language
- The second half focuses on developing Python applications
- The class is conducted in English

Rules

- Capacity is 38 students
- Do hands-on exercises on your own and make the scripts available at http://web.sfc.keio.ac.jp/~your_CNS_account/2016s-python/
- Midterm assignments and final projects must be submitted using SFC-SFS
- Attendance: NOT required
- Class cancellations, if any, will be announced via SFC-SFS

Grading

- Mid-term assignment and exercises: 30%
- Final project: 70%
- Class Participation 20% (bonus)

SFS

Please refer to SFS for

- Slides
- Exercise codes
- “Scratchpad”
- Exercise/assignments
- Solutions (case by case)
- Any other things!(annoucement, etc)

Reference

- Learning Python , O'Reilly (The Safari version is available via the Media Center)
- (Head First Python, O'Reilly)

Scripting Language

- No exact definition
- Usually interpreted, not compiled
- Cross-platform
- Unix Shell, AppleScript, Perl, Ruby, PHP, JavaScript, Python
- Scripting Language vs System Programming Language

Scripting Language (cont)

- Download the zip file from SFS
- Run the below on your terminal (one by one or on separate terminal)

```
% python indefloop.py
% gcc indefloop.c -o indefloop
Then run it ./indefloop
```

- Use “Activity Monitor” to observe the processes while they are RUNNING. Write/Submit your observation on SFS.
- (Use “Control C” to break/stop after you have finished your observation)

Python

- **a programming language that lets you work more quickly and integrate your systems more effectively**
- Open source
- Object oriented
- Translated into byte-code
- GUI

Why Python

- Software quality
- Developer productivity
- Program portability
- Support libraries
- Component integration
- Enjoyment

Python 2 vs Python 3: Quick Summary

Python 2	Python 3
`print x`	`print(x)`
`4/3 = 1`	`4/3 = 1.33333 4//3 = 1`
`raw_input()`	`input()`
`file("my_file.txt")`	`open("my_file.txt")`
`xrange(x)`	`range(x)`

Let’s start

Run Python Interpreter

1. Open Terminal
2. On the Terminal window, type: `python`

Preliminary System Checks

- Your shell must be bash

```
% ps
```

- If your shell is not bash, type

```
% bash
```

- Python version must be 2.7.x

```
% python -V
```

Prepare Your System #0

We install virtual environment because we can NOT install packages (that may be required for our class) on CNS machines. If you use your own laptop then this is not required though it is highly recommended as it would be safe to experiment without risking your system's integrity.Please check your python version though. We also will not need to install any packages in the next couple of weeks, so no rush on this. For windows users i would recommend virtual machine instead of python for windows

Download Python Setuptools

setuptools-0.6c11-py2.7.egg From

https://pypi.python.org/pypi/setuptools/0.6c11

Install Setuptools to your Python path

```
% sh setuptools-0.6c11-py2.7.egg --
prefix=~/.local
```

Prepare Your System

- Install Virtualenv

```
% ~/.local/bin/easy_install --prefix=~/.local virtualenv
```

- Create and go to the directory for this class(the below is if you're doing from CNS machine)

```
% mkdir ~/Desktop/CNS_HOMEDIR/public_html/2016s-python
```

```
% cd ~/Desktop/CNS_HOMEDIR/public_html/2016s-python
```

- Create a Python virtual environment for this class

```
% ~/.local/bin/virtualenv pythonclass
```

Prepare Your System

- Set the path for your Python

```
% mkdir -p ~/.local/lib/python2.
  7/site-packages

% export PATH=$PATH:~/.local/bin

% export PYTHONPATH=~/.
  local/lib/python2.7/site-
  packages
```

Python Virtual Environment

(ipython is optional, we 'll use "python")

```
% cd pythonclass

% source bin/activate

(to activate your virtual
environment)

% deactivate

(when you have finished)
```

Make Your Paths Permanent

- If bash is your shell, edit ~/.bash_profile ; otherwise ~/.bashrc . Add these lines:

```
export PYTHONPATH=~/.local/lib/python2.
  7/site-packages

export PATH=$PATH:~/.local/bin

export
  CLASSVIRT=~/Desktop/CNS_HOMEDIR/public_htm
  l/2015s-python/pythonclass
```

Your CNS Environment is Now Ready for This Python Class

- In bash of Terminal

```
% cd $CLASSVIRT
% source bin/activate
% python
```

Python Help Utility

- Type `help()` on the prompt
- Type `string` on the help prompt
- Press `SPACE` to show the next page
- Press `q` to quit the current help
- Type `quit` on the prompt

Numbers and Strings

```
1 + 2

1 + 2.0

4 / 3

4 / 3.0

4.0 / 3

'Hello, world!'

'Hello, ' + 'world!'
```

Comments and Unicode Strings

```
# is a comment

2.4 # this is a comment

'日本語'

u'日本語’
```

Names (Variables)

- Any letters (case sensitive)
- _
- Any numbers (not at the beginning)

Names and Assignments

```
a = 1

a

A

b = 1.0

c1 = 'python'

_d = u'日本語'

1e = 4
```

List

```
f = [ 'a', 3, a, c1 ]
g = range(10)
```

Types

```
type(a)

type(b)

type(c1)

type(_d)

type(False)

type(f)
```

Tests

```
a == 2

a == a

a == c1

c1 == _d
```

Code Blocks

```
print c1
<space>print c1
```

Indentation Defines Code Blocks

- Indenting starts a code block
- Unindenting ends a code block
- A code block must have the same indentation

Syntax: Conditional

```
In [98]: if a == b:
   ....:     print ’equal'
   ....:     <return>


In [99]: if a != b:
   ....:     print ’not equal'
   ....:     <return>
```

Conditional: if … else …

```
if a == b:
     print ’equal'
else:
   print ’not equal'
     <return>
```

For Loop

```
for i in g:
  print i, i**2
```

A Script File: helloworld.py

```
a = 'Hello, world!'
b = range(10)
for i in b:
 print i, i*'o'
 if i == 5:
  break
print a
```

A Script File

1. Edit helloworld.py; insert the below line to the file
   ```
   #!/usr/bin/env python
   ```
2. Make the file executable
   ```
   chmod +x helloworld.py
   ```
3. Run the script
   ```
   ./helloworld.py
   ```

Install Python on Your Computer

http://www.python.org/