Python for SysAdmins

Matt Schnittker and Greg Neagle
Walt Disney Animation Studios

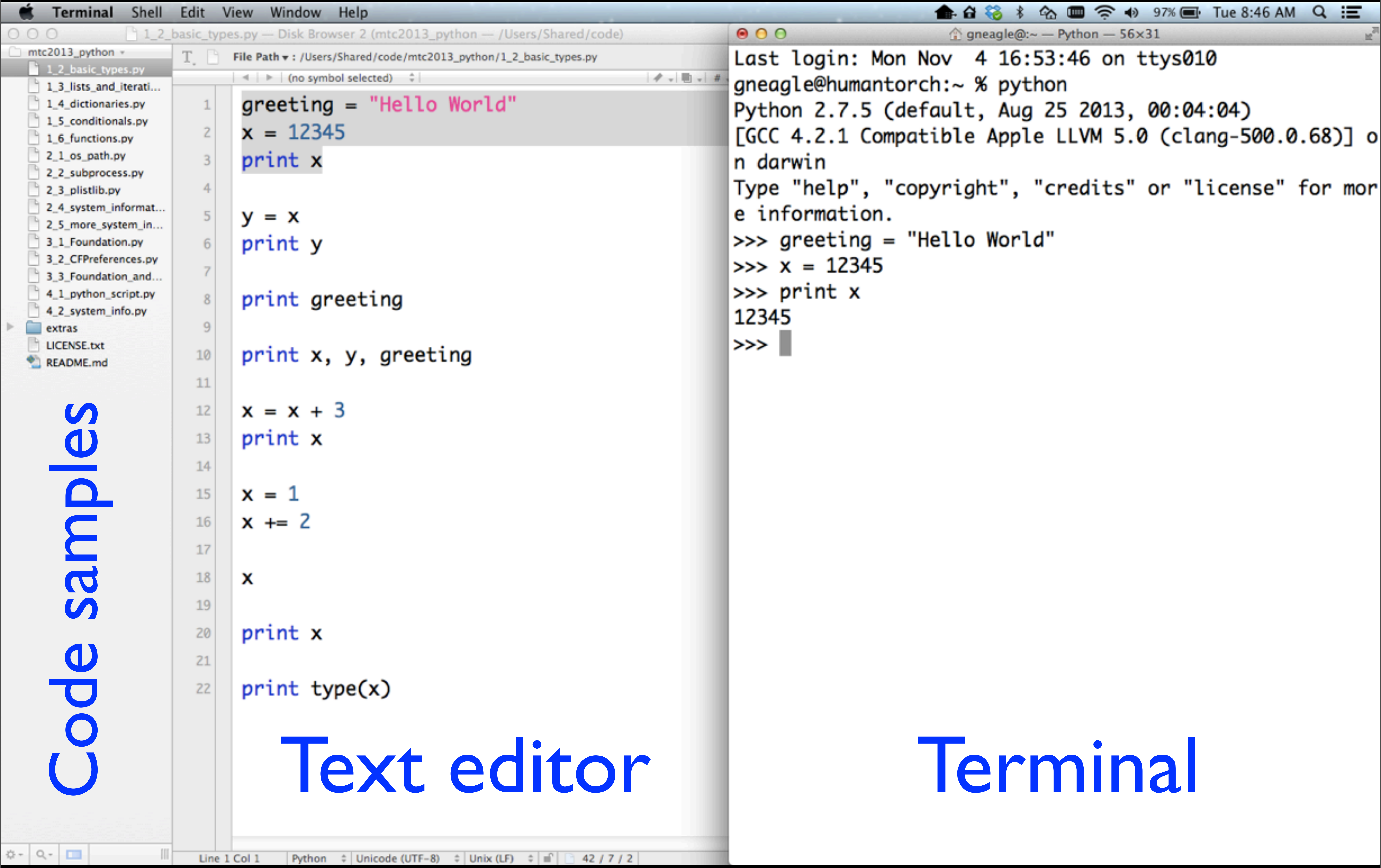
How to participate

Code samples

http://gregneagle.github.io/mtc2013_python

Multi-file text editor

TextWrangler or BBEdit or TextMate or Sublime Text



```python
greeting = "Hello World"
x = 12345
print x

y = x
print y

print greeting

print x, y, greeting

x = x + 3
print x

x = 1
x += 2

x

print x

print type(x)
```

```
Last login: Mon Nov  4 16:53:46 on ttys010
gneagle@humantorch:~ % python
Python 2.7.5 (default, Aug 25 2013, 00:04:04)
[GCC 4.2.1 Compatible Apple LLVM 5.0 (clang-500.0.68)] o
n darwin
Type "help", "copyright", "credits" or "license" for mor
e information.
>>> greeting = "Hello World"
>>> x = 12345
>>> print x
12345
>>> 
```

Code samples

Text editor

Terminal

Ready?

Starting the Python interpreter

```
% python
Python 2.7.5 (default, Aug 25 2013,
00:04:04)
[GCC 4.2.1 Compatible Apple LLVM 5.0
(clang-500.0.68)] on darwin
Type "help", "copyright", "credits" or
"license" for more information.
>>>
```

Basic Types: string, int

`1_2_basic_types` .py

```
>>> greeting = "Hello World"
>>> x = 12345
>>> print x
12345
>>> y = x
>>> print y
12345
>>> print greeting
Hello World
>>> print x, y, greeting
12345 12345 Hello World
```

1_2_basic_types.py

```
>>> x = x + 3
>>> print x
12348
>>> x = 1
>>> x += 2
>>> x
3
>>> print x
3
print type(x)
<type 'int'>
```

1_2_basic_types.py

Basic Types: lists and iteration

`1_3_lists_and_iteration`.py

```
>>> my_list = [1, 2, 3, 4, 56]
>>> my_list[0]
1
>>> my_list[-1]
56
>>> my_list[1:3]
[2, 3]
```

1_3_lists_and_iteration .py

```
>>> my_list = [1, 2, 3, 4, 56]
>>> for item in my_list:
...     print item
...
1
2
3
4
56
```

This space is important!

1_3_lists_and_iteration .py

Seque: Python and white space

(Hint: it's really important.)

```
#!/bin/sh
for NUM in 1 2 3 4 5 ; do
  echo "$NUM"
done
echo "Done!"
```

```
#!/bin/sh
for NUM in 1 2 3 4 5
do
echo "$NUM"
done
echo "Done!"
```

```
#!/bin/sh
for NUM in 1 2 3 4 5 ; do echo "$NUM" ; done ; echo "Done!"
```

```
#!/usr/bin/perl
foreach $num (1, 2, 3, 4, 5) {
    print "$num\n";
}
print "Done!\n";
```

```
#!/usr/bin/perl
foreach $num (1, 2, 3, 4, 5)
{
print "$num\n";
}
print "Done!\n";
```

```
#!/usr/bin/perl
foreach $num (1, 2, 3, 4, 5) { print "$num\n"; } print "Done!\n";
```

```
#!/usr/bin/python

for NUM in range(1, 6):
    print NUM
print "Done!"
```

```
#!/usr/bin/python

for NUM in range(1, 6):
    print NUM
    if NUM == 3:
        print "I just printed 3!"
        print "Hope you liked that."
    elif NUM == 4:
        print "I just printed 4."
    print
print "Done!"
```

now, where were we?

`1_3_lists_and_iteration` .py

```
>>> stuff = [4, "birds", 3.14, True]
>>> for thing in stuff:
...     print thing, type(thing)
...
4 <type 'int'>
birds <type 'str'>
3.14 <type 'float'>
True <type 'bool'>
```

1_3_lists_and_iteration.py

Basic Types: dictionaries

`1_4_dictionaries` .py

```
>>> student_ids = {"Bob": 12334546,
      "Sally": 23666, "Jane": 45678899}
>>> student_ids["Jane"]
45678899

>>> for key in student_ids:
...     print student_ids[key]
...
45678899
12334546
23666
```

1_4_dictionaries.py

```
>>> for key in student_ids:
...     print key, student_ids[key]
...
Jane 45678899
Bob 12334546
Sally 23666
```

1_4_dictionaries.py

```
>>> prefs = {"UseColor": False,
         "NumberOfCopies": 2,
         "HeaderText": "Property of me"}
>>> print prefs["HeaderText"]
Property of me
```

```
<plist version="1.0">
<dict>
   <key>HeaderText</key>
   <string>Property of me</string>
   <key>NumberOfCopies</key>
   <integer>2</integer>
   <key>UseColor</key>
   <false/>
</dict>
</plist>
```

1_4_dictionaries.py

Conditionals

`1_5_conditionals`.py

```
>>> my_grade = "A"
>>> if my_grade == "A":
...     print "Party More"
... else:
...     print "You partied enough"
...
Party More
```

1_5_conditionals.py

```
>>> my_grade = "B"
>>> if my_grade == "A":
...     print "Party More"
... elif my_grade == "B":
...     print "You partied enough"
... else:
...     print "You partied too much!"
...
You partied enough
```

1_5_conditionals.py

Functions

`1_6_functions.py`

```
>>> def question(question_number):
...     if question_number == 1:
...         return "What is your name?"
...     elif question_number == 2:
...         return "What is your quest?"
...     elif question_number == 3:
...         return "What is your favorite color?"
...     else:
...         return "Ahhhhhh"
...
>>> print question(1)
What is your name?

>>> print question(3)
What is your favorite color?
```

1_6_functions .py

```
>>> def header(text, level):
...     open_tag = "<h" + str(level) + ">"
...     close_tag = "</h" + str(level) + ">"
...     return open_tag + text + close_tag
...
>>> print header("My Great Title", 1)
<h1>My Great Title</h1>

>>> print header("My Subtitle", 2)
<h2>My Subtitle</h2>
```

1_6_functions.py

```
>>> def header(text, level=1):
...     open_tag = "<h" + str(level) + ">"
...     close_tag = "</h" + str(level) + ">"
...     return open_tag + text + close_tag
...
>>> print header("My Great Title")
<h1>My Great Title</h1>
```

1_6_functions .py

Standard Library Modules

os, os.path

2_1_os_path.py

```
>>> import os
>>> PATH = "/Library/Preferences/com.apple.SoftwareUpdate.plist"
>>> os.path.exists(PATH)
True

>>> os.path.isdir(PATH)
False

>>> os.path.dirname(PATH)
'/Library/Preferences'

>>> os.path.basename(PATH)
'com.apple.SoftwareUpdate.plist'
```

2_1_os_path .py

```
>>> home = os.path.expanduser("~")
>>> print home
/Users/gneagle

>>> prefs_dir = os.path.join(home, "Library/Preferences")
>>> print prefs_dir
/Users/gneagle/Library/Preferences

>>> for filename in os.listdir(prefs_dir):
...     print filename
...
.DS_Store
.GlobalPreferences.plist
Adobe
Adobe Photoshop CS6 Paths
...etc...
```

2_1_os_path.py

Running external commands

`2_2_subprocess.py`

```
>>> import subprocess
>>> cmd = ["/usr/bin/open", "http://disneyanimation.com"]
>>> subprocess.call(cmd)
0
```

2_2_subprocess .py

```
>>> cmd = ['/usr/sbin/pkgutil', '--pkgs']
>>> result = subprocess.call(cmd)
com.adobe.pkg.AIR
com.adobe.pkg.FlashPlayer
com.apple.pkg.AdditionalEssentials
com.apple.pkg.AdditionalSpeechVoices
...etc...

>>> print result
0
```

2_2_subprocess.py

```
>>> cmd = ['/usr/sbin/pkgutil', '--pkgs']
>>> proc = subprocess.Popen(cmd,
stdout=subprocess.PIPE, stderr=subprocess.PIPE)
>>> (output, error_output) = proc.communicate()

>>> print "Output:", output
Output: com.adobe.pkg.AIR
com.adobe.pkg.FlashPlayer
com.amazon.Kindle
com.apple.pkg.AdditionalEssentials
...etc..
>>> print "Error output:", error_output
Error output:
>>> print "Return code:", proc.returncode
Return code: 0
```

2_2_subprocess.py

```
>>> print "Output:", output
Output: com.adobe.pkg.AIR
com.adobe.pkg.FlashPlayer
com.amazon.Kindle
com.apple.pkg.AdditionalEssentials
...etc..
>>> print "Error output:", error_output
Error output:
>>> print "Return code:", proc.returncode
Return code: 0
```

2_2_subprocess.py

```
# don't run this one!
subprocess.call(['/sbin/shutdown', '-r', 'now'])
```

2_2_subprocess .py

Working with Plists

`2_3_plistlib.py`

```
>>> import plistlib
>>> filename = "/Applications/Safari.app/Contents/Info.plist"
>>> info = plistlib.readPlist(filename)
>>> print info["CFBundleGetInfoString"]
7.0, Copyright © 2003-2013 Apple Inc.

>>> version = info["CFBundleShortVersionString"]
>>> print version
7.0
```

2_3_plistlib.py

Complex values

`2_3_plistlib` .py

info["CFBundleURLTypes"]

```
<key>CFBundleURLTypes</key>
<array>
    <dict>
        <key>CFBundleURLName</key>
        <string>Web site URL</string>
        <key>CFBundleURLSchemes</key>
        <array>
            <string>http</string>
            <string>https</string>
        </array>
        <key>LSIsAppleDefaultForScheme</key>
        <true/>
    </dict>
    <dict>
        <key>CFBundleURLName</key>
        <string>Local file URL</string>
        <key>CFBundleURLSchemes</key>
        <array>
            <string>file</string>
        </array>
    </dict>
</array>
```

2_3_plistlib.py

info["CFBundleURLTypes"][0]

```
    <key>CFBundleURLTypes</key>
    <array>
        <dict>
            <key>CFBundleURLName</key>
            <string>Web site URL</string>
            <key>CFBundleURLSchemes</key>
            <array>
                <string>http</string>
                <string>https</string>
            </array>
            <key>LSIsAppleDefaultForScheme</key>
            <true/>
        </dict>
        <dict>
            <key>CFBundleURLName</key>
            <string>Local file URL</string>
            <key>CFBundleURLSchemes</key>
            <array>
                <string>file</string>
            </array>
        </dict>
    </array>
```

2_3_plistlib.py

info["CFBundleURLTypes"][0]["CFBundleURLSchemes"]

```
<key>CFBundleURLTypes</key>
<array>
    <dict>
        <key>CFBundleURLName</key>
        <string>Web site URL</string>
        <key>CFBundleURLSchemes</key>
        <array>
            <string>http</string>
            <string>https</string>
        </array>
        <key>LSIsAppleDefaultForScheme</key>
        <true/>
    </dict>
    <dict>
        <key>CFBundleURLName</key>
        <string>Local file URL</string>
        <key>CFBundleURLSchemes</key>
        <array>
            <string>file</string>
        </array>
    </dict>
</array>
```

2_3_plistlib.py

info["CFBundleURLTypes"][0]["CFBundleURLSchemes"][0]

```
<key>CFBundleURLTypes</key>
<array>
    <dict>
        <key>CFBundleURLName</key>
        <string>Web site URL</string>
        <key>CFBundleURLSchemes</key>
        <array>
            <string>http</string>
            <string>https</string>
        </array>
        <key>LSIsAppleDefaultForScheme</key>
        <true/>
    </dict>
    <dict>
        <key>CFBundleURLName</key>
        <string>Local file URL</string>
        <key>CFBundleURLSchemes</key>
        <array>
            <string>file</string>
        </array>
    </dict>
</array>
```

2_3_plistlib .py

```
>>> print info["CFBundleURLTypes"][0]["CFBundleURLSchemes"][0]
http
```

2_3_plistlib .py

plistlib caveat

```
>>> filename = "/Library/Preferences/com.apple.loginwindow.plist"
>>> plistinfo = plistlib.readPlist(filename)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/System/Library/Frameworks/Python.framework/Versions/2.7/
lib/python2.7/plistlib.py", line 78, in readPlist
    rootObject = p.parse(pathOrFile)
  File "/System/Library/Frameworks/Python.framework/Versions/2.7/
lib/python2.7/plistlib.py", line 406, in parse
    parser.ParseFile(fileobj)
xml.parsers.expat.ExpatError: not well-formed (invalid token):
line 1, column 8
```

2_3_plistlib.py

Getting system information

`2_4_system_information`.py

```
>>> import sys
>>> sys.platform
'darwin'
```

2_4_system_information.py

```
>>> import os
>>> os.uname()
('Darwin', 'gregs-mac.example.com',
'13.0.0', 'Darwin Kernel Version 13.0.0:
Thu Sep 19 22:22:27 PDT 2013;
root:xnu-2422.1.72~6/RELEASE_X86_64',
'x86_64')

>>> os.uname()[2]
'13.0.0'
>>> os.uname()[4]
'x86_64'
```

2_4_system_information.py

```
>>> import platform
>>> platform.mac_ver()
('10.8.5', ('', '', ''), 'x86_64')
>>> platform.mac_ver()[0]
'10.8.5'
```

2_4_system_information.py

```
>>> import subprocess
>>> cmd = ['/usr/bin/sw_vers', '-productVersion']
>>> subprocess.check_output(cmd)
'10.9\n'

>>> subprocess.check_output(cmd).strip()
'10.9'
```

2_4_system_information .py

Getting even more system information

`2_5_more_system_information` .py

```
>>> import subprocess
>>> cmd = ['/usr/sbin/system_profiler',
           'SPHardwareDataType', '-xml']
>>> proc = subprocess.call(cmd)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<array>
   <dict>
      <...gi-normous plist details go here...>
   </dict>
</array>
</plist>
```

2_5_more_system_information.py

```
>>> proc = subprocess.Popen(cmd,
      stdout=subprocess.PIPE, stderr=subprocess.PIPE)
>>> (output, error) = proc.communicate()
>>> print output
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<array>
   <dict>
      <...gi-normous plist details go here...>
   </dict>
</array>
</plist>
```

2_5_more_system_information .py

```
>>> import plistlib
>>> info = plistlib.readPlistFromString(output)
>>> info
[{'_parentDataType': 'SPRootDataType', '_timeStamp':
datetime.datetime(2013, 10, 19, 22, 26, 34),
'_detailLevel': '-2', '_dataType':
'SPHardwareDataType', '_versionInfo':
{'com.apple.SystemProfiler.SPPlatformReporter':
'1440'}, '_properties': {'platform_product_name':
{'_order': '2'}, 'machine_name': {'_order': '10'},
'l3_cache_size': {'_order': '27'},
'SMC_version_other': {'_order': '82'},
'riser_serial_number': {'_detailLevel': '0',
'_order': '92'}, 'minimum_processor_speed':
{'_order': '18'}, 'cpu_interconnect_speed':
{'_order': '46'}, 'LOM_revision': {'_order': '85'},
'machine_model': {'_order': '11'},
'SMC_version_riser': {'_order': '81'}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<array>
    <dict>
        <key>_SPCommandLineArguments</key>
        <array>
            <string>/usr/sbin/system_profiler</string>
            <string>-nospawn</string>
            <string>-xml</string>
            <string>SPHardwareDataType</string>
            <string>-detailLevel</string>
            <string>full</string>
        </array>
        <key>_SPCompletionInterval</key>
        <real>0.013341963291168213</real>
        <key>_SPResponseTime</key>
        <real>0.13739603757858276</real>
        <key>_dataType</key>
        <string>SPHardwareDataType</string>
        <key>_detailLevel</key>
        <string>-2</string>
        <key>_items</key>
        <array>
            <dict>
                <key>SMC_version_system</key>
                <string>2.13f7</string>
                <key>_name</key>
                <string>hardware_overview</string>
                <key>boot_rom_version</key>
                <string>MBA61.0099.B04</string>
                <key>cpu_type</key>
                <string>Intel Core i7</string>
```

info

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<array>
    <dict>
        <key>_SPCommandLineArguments</key>
        <array>
            <string>/usr/sbin/system_profiler</string>
            <string>-nospawn</string>
            <string>-xml</string>
            <string>SPHardwareDataType</string>
            <string>-detailLevel</string>
            <string>full</string>
        </array>
        <key>_SPCompletionInterval</key>
        <real>0.013341963291168213</real>
        <key>_SPResponseTime</key>
        <real>0.13739603757858276</real>
        <key>_dataType</key>
        <string>SPHardwareDataType</string>
        <key>_detailLevel</key>
        <string>-2</string>
        <key>_items</key>
        <array>
            <dict>
                <key>SMC_version_system</key>
                <string>2.13f7</string>
                <key>_name</key>
                <string>hardware_overview</string>
                <key>boot_rom_version</key>
                <string>MBA61.0099.B04</string>
                <key>cpu_type</key>
                <string>Intel Core i7</string>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<array>
    <dict>
        <key>_SPCommandLineArguments</key>
        <array>
            <string>/usr/sbin/system_profiler</string>
            <string>-nospawn</string>
            <string>-xml</string>
            <string>SPHardwareDataType</string>
            <string>-detailLevel</string>
            <string>full</string>
        </array>
        <key>_SPCompletionInterval</key>
        <real>0.013341963291168213</real>
        <key>_SPResponseTime</key>
        <real>0.13739603757858276</real>
        <key>_dataType</key>
        <string>SPHardwareDataType</string>
        <key>_detailLevel</key>
        <string>-2</string>
        <key>_items</key>
        <array>
            <dict>
                <key>SMC_version_system</key>
                <string>2.13f7</string>
                <key>_name</key>
                <string>hardware_overview</string>
                <key>boot_rom_version</key>
                <string>MBA61.0099.B04</string>
                <key>cpu_type</key>
                <string>Intel Core i7</string>
```

info[0]

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<array>
    <dict>
        <key>_SPCommandLineArguments</key>
        <array>
            <string>/usr/sbin/system_profiler</string>
            <string>-nospawn</string>
            <string>-xml</string>
            <string>SPHardwareDataType</string>
            <string>-detailLevel</string>
            <string>full</string>
        </array>
        <key>_SPCompletionInterval</key>
        <real>0.013341963291168213</real>
        <key>_SPResponseTime</key>
        <real>0.13739603757858276</real>
        <key>_dataType</key>
        <string>SPHardwareDataType</string>
        <key>_detailLevel</key>
        <string>-2</string>
        <key>_items</key>
        <array>
            <dict>
                <key>SMC_version_system</key>
                <string>2.13f7</string>
                <key>_name</key>
                <string>hardware_overview</string>
                <key>boot_rom_version</key>
                <string>MBA61.0099.B04</string>
                <key>cpu_type</key>
                <string>Intel Core i7</string>
```

info[0]['_items']

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<array>
    <dict>
        <key>_SPCommandLineArguments</key>
        <array>
            <string>/usr/sbin/system_profiler</string>
            <string>-nospawn</string>
            <string>-xml</string>
            <string>SPHardwareDataType</string>
            <string>-detailLevel</string>
            <string>full</string>
        </array>
        <key>_SPCompletionInterval</key>
        <real>0.013341963291168213</real>
        <key>_SPResponseTime</key>
        <real>0.13739603757858276</real>
        <key>_dataType</key>
        <string>SPHardwareDataType</string>
        <key>_detailLevel</key>
        <string>-2</string>
        <key>_items</key>
        <array>
            <dict>
                <key>SMC_version_system</key>
                <string>2.13f7</string>
                <key>_name</key>
                <string>hardware_overview</string>
                <key>boot_rom_version</key>
                <string>MBA61.0099.B04</string>
                <key>cpu_type</key>
                <string>Intel Core i7</string>
```

info[0]['_items'][0]

```
>>> hardware_info = info[0]['_items'][0]
>>> hardware_info.keys()
['platform_UUID',
'current_processor_speed',
'machine_name', 'l2_cache_core',
'SMC_version_system', 'physical_memory',
'number_processors', '_name',
'machine_model', 'cpu_type',
'serial_number', 'boot_rom_version',
'packages', 'l3_cache']
```

2_5_more_system_information.py

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<array>
    <dict>
        <key>_SPCommandLineArguments</key>
        <array>
            <string>/usr/sbin/system_profiler</string>
            <string>-nospawn</string>
            <string>-xml</string>
            <string>SPHardwareDataType</string>
            <string>-detailLevel</string>
            <string>full</string>
        </array>
        <key>_SPCompletionInterval</key>
        <real>0.013341963291168213</real>
        <key>_SPResponseTime</key>
        <real>0.13739603757858276</real>
        <key>_dataType</key>
        <string>SPHardwareDataType</string>
        <key>_detailLevel</key>
        <string>-2</string>
        <key>_items</key>
        <array>
            <dict>
                <key>SMC_version_system</key>
                <string>2.13f7</string>
                <key>_name</key>
                <string>hardware_overview</string>
                <key>boot_rom_version</key>
                <string>MBA61.0099.B04</string>
                <key>cpu_type</key>
                <string>Intel Core i7</string>
                <key>current_processor_speed</key>
```

hardware_info

hardware_info

```
<string>-2</string>
<key>_items</key>
<array>
    <dict>
        <key>SMC_version_system</key>
        <string>2.13f7</string>
        <key>_name</key>
        <string>hardware_overview</string>
        <key>boot_rom_version</key>
        <string>MBA61.0099.B04</string>
        <key>cpu_type</key>
        <string>Intel Core i7</string>
        <key>current_processor_speed</key>
        <string>1.7 GHz</string>
        <key>l2_cache_core</key>
        <string>256 KB</string>
        <key>l3_cache</key>
        <string>4 MB</string>
        <key>machine_model</key>
        <string>MacBookAir6,2</string>
        <key>machine_name</key>
        <string>MacBook Air</string>
        <key>number_processors</key>
        <integer>2</integer>
        <key>packages</key>
        <integer>1</integer>
        <key>physical_memory</key>
        <string>8 GB</string>
        <key>platform_UUID</key>
        <string>F89FF644-ED21-5FDC-9DBC-D80A25A1F6DC</string>
        <key>serial_number</key>
        <string>C02KX5KXF6T6</string>
    </dict>
</array>
```

```
>>> hardware_info['cpu_type']
'Intel Core i7'

>>> hardware_info['machine_model']
'MacBookAir6,2'

>>> hardware_info['serial_number']
'C02KX5KXF6T6'
```

2_5_more_system_information.py

```
>>> for key, value in hardware_info.items():
...     print print str(key) + ": " + str(value)
...
platform_UUID: F89FF644-ED21-5FDC-9DBC-D80A25A1F6DC
current_processor_speed: 1.7 GHz
machine_name: MacBook Air
l2_cache_core: 256 KB
SMC_version_system: 2.13f7
physical_memory: 8 GB
number_processors: 2
_name: hardware_overview
machine_model: MacBookAir6,2
cpu_type: Intel Core i7
serial_number: C02KX5KXF6T6
boot_rom_version: MBA61.0099.B04
packages: 1
l3_cache: 4 MB
```

2_5_more_system_information.py

PyObjC

Foundation functions

`3_1_Foundation` .py

```
>>> import Foundation
>>> Foundation.NSUserName()
u'gneagle'
>>> Foundation.NSFullUserName()
u'Greg Neagle'
>>> Foundation.NSHomeDirectory()
u'/Users/gneagle'
```

https://developer.apple.com/library/mac/documentation/cocoa/reference/foundation/Miscellaneous/Foundation_Functions/Reference/reference.html

`3_1_Foundation` .py

CFPreferences

`3_2_CFPreferences.py`

```
>>> import CoreFoundation
>>> print CoreFoundation.CFPreferencesCopyAppValue(
                      "HomePage", "com.apple.Safari")
http://www.google.com/

>>> new_home = "http://disneyanimation.com"
>>> CoreFoundation.CFPreferencesSetAppValue(
            "HomePage", new_home, "com.apple.Safari")

>>> print CoreFoundation.CFPreferencesCopyAppValue(
                      "HomePage", "com.apple.Safari")
http://disneyanimation.com
```

3_2_CFPreferences.py

```
<key>ManagedPlugInPolicies</key>
<dict>
    <key>com.macromedia.Flash Player.plugin</key>
    <dict>
        <key>PlugInDisallowPromptBeforeUseDialog</key>
        <true/>
        <key>PlugInFirstVisitPolicy</key>
        <string>PlugInPolicyAllowWithSecurityRestrictions</string>
     </dict>
</dict>
```

```
>>> my_policy = {
...     "com.macromedia.Flash Player.plugin": {
...         "PlugInDisallowPromptBeforeUseDialog":
                                                True,
...         "PlugInFirstVisitPolicy":
             "PlugInPolicyAllowWithSecurityRestrictions",
...     },
... }

>>> CoreFoundation.CFPreferencesSetAppValue(
  "ManagedPlugInPolicies", my_policy, "com.apple.Safari")
```

https://developer.apple.com/library/mac/documentation/CoreFoundation/Reference/CFPreferencesUtils/Reference/reference.html

plists with Foundation

`3_3_Foundation_plists` .py

```
>>> from Foundation import NSData
>>> from Foundation import NSPropertyListSerialization
>>> from Foundation import NSPropertyListMutableContainersAndLeaves

>>> filename = "/Library/Preferences/com.apple.loginwindow.plist"
>>> plist_data = NSData.dataWithContentsOfFile_(filename)
>>> (dataObject, plistFormat, error) = (
...     NSPropertyListSerialization.propertyListWithData_options_format_error_(
...         plist_data, NSPropertyListMutableContainersAndLeaves, None, None))

>>> print dataObject
{
    DesktopPicture = "/Library/Desktop Pictures/Disney/Bambi.jpg";
    MCXLaunchAfterUserLogin = 1;
    MCXLaunchOnUserLogout =     {
        gneagle = 1;
    };
    OptimizerLastRunForBuild = 27282272;
    OptimizerLastRunForSystem = 168361984;
    RetriesUntilHint = 0;
    SHOWFULLNAME = 1;
    lastLoginPanic = "404536730.80456";
    lastUser = loggedIn;
    lastUserName = gneagle;
}
```

3_3_Foundation_plists.py

```
>>> dataObject.keys()
(
    LoginHook,
    OptimizerLastRunForSystem,
    RetriesUntilHint,
    LogoutHook,
    lastLoginPanic,
    DesktopPicture,
    OptimizerLastRunForBuild,
    lastUser,
    MCXLaunchAfterUserLogin,
    lastUserName,
    MCXLaunchOnUserLogout,
    SHOWFULLNAME
)

>>> dataObject['lastUserName']
u'gneagle'
>>> dataObject['SHOWFULLNAME']
True
```

3_3_Foundation_plists .py

https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/PropertyLists/Introduction/Introduction.html

Python scripts

```
#!/usr/bin/python

"""Calls system_profiler and prints hardware info about
the current machine"""

import plistlib
import subprocess

cmd = ['/usr/sbin/system_profiler', 'SPHardwareDataType', '-xml']
proc = subprocess.Popen(cmd, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
(output, error) = proc.communicate()

info = plistlib.readPlistFromString(output)

hardware_info = info[0]['_items'][0]
for key, value in hardware_info.items():
    print str(key) + ": " + str(value)
```

“Sh-bang” line

4_1_python_script.py

```
#!/usr/bin/python

"""Calls system_profiler and prints hardware info about
the current machine"""

import plistlib
import subprocess

cmd = ['/usr/sbin/system_profiler', 'SPHardwareDataType', '-xml']
proc = subprocess.Popen(cmd, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
(output, error) = proc.communicate()

info = plistlib.readPlistFromString(output)

hardware_info = info[0]['_items'][0]
for key, value in hardware_info.items():
    print str(key) + ": " + str(value)
```

doc string

4_1_python_script .py

```
#!/usr/bin/python

"""Calls system_profiler and prints hardware info about
the current machine"""

import plistlib
import subprocess

cmd = ['/usr/sbin/system_profiler', 'SPHardwareDataType', '-xml']
proc = subprocess.Popen(cmd, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
(output, error) = proc.communicate()

info = plistlib.readPlistFromString(output)

hardware_info = info[0]['_items'][0]
for key, value in hardware_info.items():
    print str(key) + ": " + str(value)
```

imports

4_1_python_script.py

```
#!/usr/bin/python

"""Calls system_profiler and prints hardware info about
the current machine"""

import plistlib
import subprocess

cmd = ['/usr/sbin/system_profiler', 'SPHardwareDataType', '-xml']
proc = subprocess.Popen(cmd, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
(output, error) = proc.communicate()

info = plistlib.readPlistFromString(output)

hardware_info = info[0]['_items'][0]
for key, value in hardware_info.items():
    print str(key) + ": " + str(value)
```

“main”

4_1_python_script .py

Better formatting

`4_2_system_info` .py

```
#!/usr/bin/python

"""Calls system_profiler and prints hardware info about
the current machine"""

import plistlib
import subprocess

def main():
    cmd = ['/usr/sbin/system_profiler', 'SPHardwareDataType', '-xml']
    proc = subprocess.Popen(cmd, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    (output, error) = proc.communicate()

    info = plistlib.readPlistFromString(output)

    hardware_info = info[0]['_items'][0]
    for key, value in hardware_info.items():
        print str(key) + ": " + str(value)

if __name__ == "__main__":
    main()
```

main

4_2_system_info.py

```
#!/usr/bin/python

"""Calls system_profiler and prints hardware info about
the current machine"""

import plistlib
import subprocess

def main():
    cmd = ['/usr/sbin/system_profiler', 'SPHardwareDataType', '-xml']
    proc = subprocess.Popen(cmd, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    (output, error) = proc.communicate()

    info = plistlib.readPlistFromString(output)

    hardware_info = info[0]['_items'][0]
    for key, value in hardware_info.items():
        print str(key) + ": " + str(value)

if __name__ == "__main__":
    main()
```

calls the main function

4_2_system_info .py

```
% python 4_2_system_info.py
platform_UUID: F89FF644-ED21-5FDC-9DBC-D80A25A1F6DC
current_processor_speed: 1.7 GHz
machine_name: MacBook Air
l2_cache_core: 256 KB
SMC_version_system: 2.13f7
physical_memory: 8 GB
number_processors: 2
_name: hardware_overview
machine_model: MacBookAir6,2
cpu_type: Intel Core i7
serial_number: C02KX5KXF6T6
boot_rom_version: MBA61.0099.B04
packages: 1
l3_cache: 4 MB
```

4_2_system_info .py

More scripts

Check out the extras folder:

`01_set_desktop_picture.py:`

- Uses AppKit methods to set desktop picture

`02_mirrortool.py:`

- Uses Quartz/CoreGraphics methods to configure display mirroring

`FoundationPlist.py:`

- A complete set of functions for reading and writing plists using Cocoa Foundation methods

Learn more

http://www.python.org/

http://www.diveintopython.net

MacTech Magazine 2013: MacEnterprise
- lots of PyObjC examples; GUI programming, too

https://www.coursera.org/course/interactivepython