

Python and macOS



Greg Neagle, Walt Disney Animation Studios

Python Planning



Greg Neagle, Walt Disney Animation Studios

Snakes on a Plan



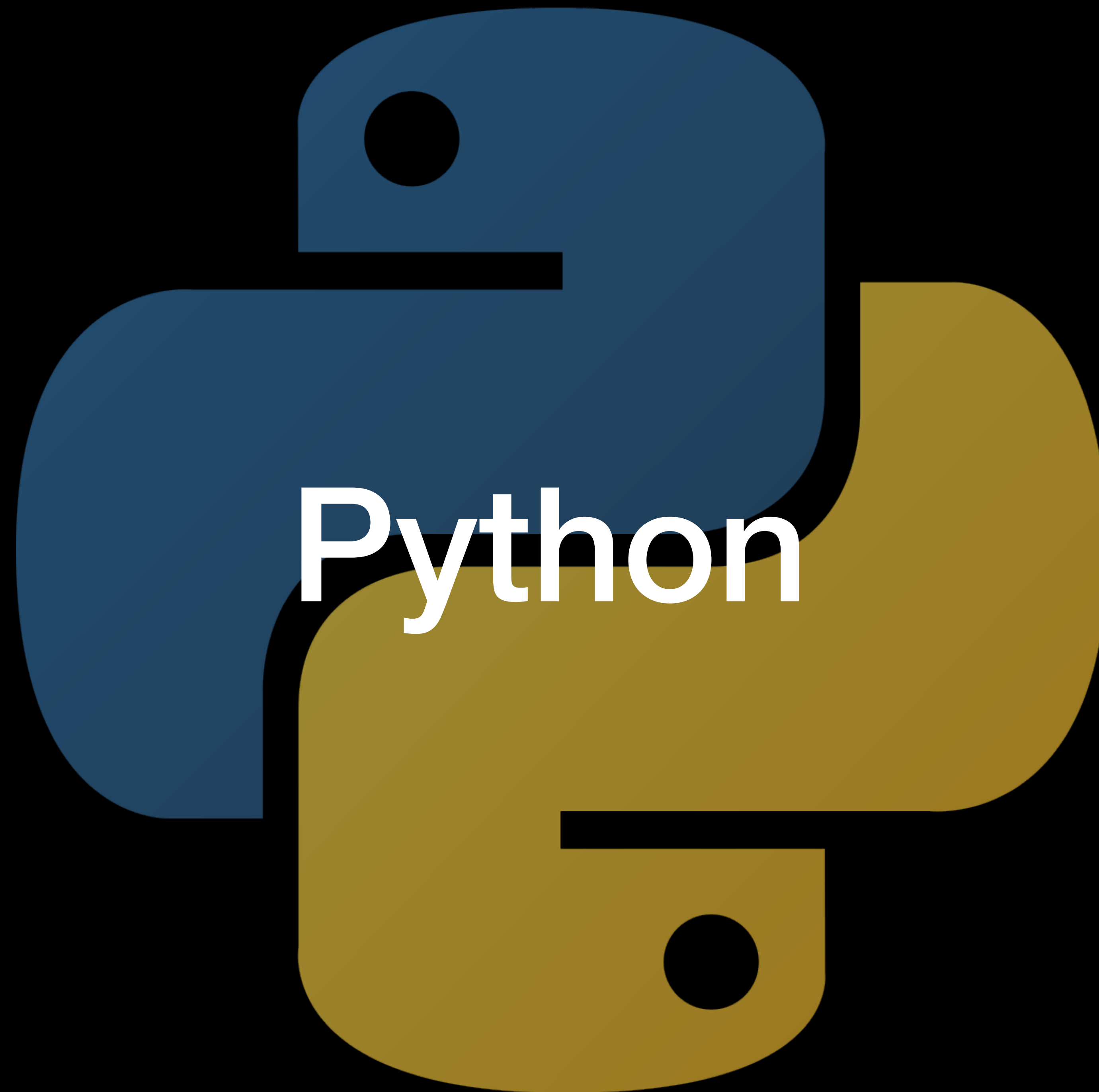
Greg Neagle, Walt Disney Animation Studios

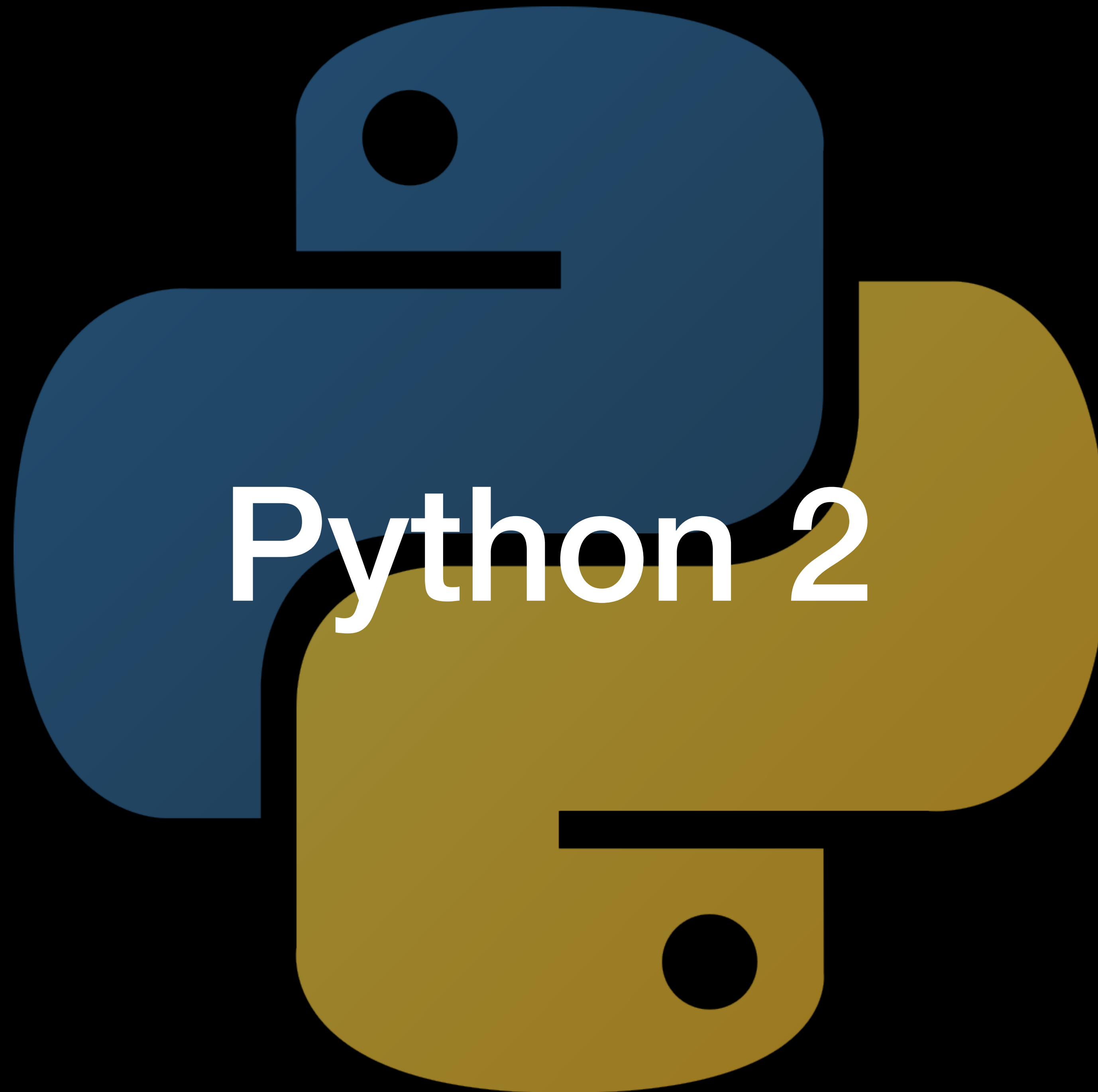
Snakes on a Plan:

Python, Apple, and You



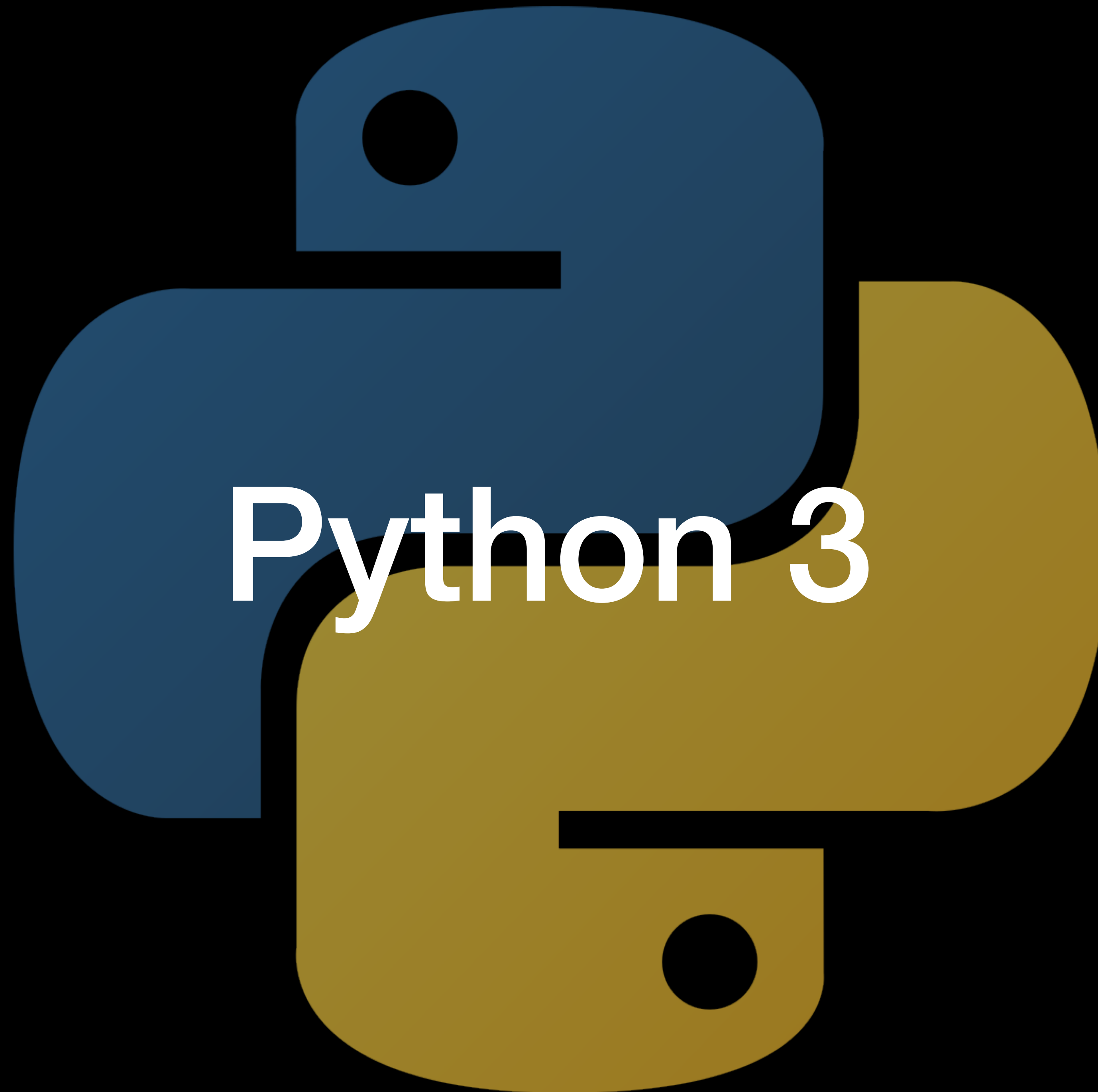
Greg Neagle, Walt Disney Animation Studios





The image features the Python logo, which consists of two interlocking snakes. The left snake is blue and the right snake is yellow. They are positioned behind the text. The text "Python 2 is EOL" is written in a white, sans-serif font, centered horizontally across the middle of the image.

Python 2 is EOL





Scripting Language Runtimes

Deprecations

- Scripting language runtimes such as Python, Ruby, and Perl are included in macOS for compatibility with legacy software. Future versions of macOS won't include scripting language runtimes by default, and might require you to install additional packages. If your software depends on scripting languages, it's recommended that you bundle the runtime within the app. (49764202)
- Use of Python 2.7 isn't recommended as this version is included in macOS for compatibility with legacy software. Future versions of macOS won't include Python 2.7. Instead, it's recommended that you run `python3` from within Terminal. (51097165)

Catalina

```
bash-3.2$ sw_vers  
ProductName: Mac OS X  
ProductVersion: 10.15.6  
BuildVersion: 19G2021
```

```
bash-3.2$ python
```

WARNING: Python 2.7 is not recommended.

This version is included in macOS for compatibility with legacy software.

Future versions of macOS will not include Python 2.7.

Instead, it is recommended that you transition to using 'python3' from within Terminal.

Python 2.7.16 (default, Jun 5 2020, 22:59:21)

[GCC 4.2.1 Compatible Apple LLVM 11.0.3 (clang-1103.0.29.20) (-macos10.15-objc- on darwin

Type "help", "copyright", "credits" or "license" for more information.

```
>>>
```

Big Sur

```
bash-3.2$ sw_vers  
ProductName: macOS  
ProductVersion: 11.0  
BuildVersion: 20A5364e
```

```
bash-3.2$ python
```

WARNING: Python 2.7 is not recommended.

This version is included in macOS for compatibility with legacy software.

Future versions of macOS will not include Python 2.7.

Instead, it is recommended that you transition to using 'python3' from within Terminal.

Python 2.7.16 (default, Aug 24 2020, 12:22:49)

[GCC Apple LLVM 12.0.0 (clang-1200.0.30.1) [+internal-os, ptrauth=sign+stri on darwin

Type "help", "copyright", "credits" or "license" for more information.

```
>>>
```

Big Sur

```
bash-3.2$ sw_vers  
ProductName: macOS  
ProductVersion: 11.0  
BuildVersion: 20A5364e
```

```
bash-3.2$ python
```

```
WARNING: Python 2.7 is not recommended.  
This version is included in macOS for compatibility with legacy software.  
Future versions of macOS will not include Python 2.7.  
Instead, it is recommended that you transition to using 'python3' from within Terminal.
```

```
Python 2.7.16 (default, Aug 24 2020, 12:22:49)  
[GCC Apple LLVM 12.0.0 (clang-1200.0.30.1) [+internal-os, ptrauth=sign+stri on darwin  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import OpenDirectory
```


Big Sur

```
bash-3.2$ sw_vers
ProductName: macOS
ProductVersion: 11.0
BuildVersion: 20A5364e
```

```
bash-3.2$ python
```

```
WARNING: Python 2.7 is not recommended.
This version is included in macOS for compatibility with legacy software.
Future versions of macOS will not include Python 2.7.
Instead, it is recommended that you transition to using 'python3' from within Terminal.
```

```
Python 2.7.16 (default, Aug 24 2020, 12:22:49)
[GCC Apple LLVM 12.0.0 (clang-1200.0.30.1) [+internal-os, ptrauth-isa=sign+stri on darwin
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import OpenDirectory
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
  File "/System/Library/Frameworks/Python.framework/Versions/2.7/Extras/lib/python/PyObjC/
OpenDirectory/__init__.py", line 12, in <module>
```

```
    objc.pathForFramework("/System/Library/Frameworks/OpenDirectory.framework/Frameworks/
OpenDirectory.framework"),
```

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import OpenDirectory
```

Big Sur

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

File "/System/Library/Frameworks/Python.framework/Versions/2.7/Extras/lib/python/PyObjC/OpenDirectory/__init__.py", line 12, in <module>

objc.pathForFramework("/System/Library/Frameworks/OpenDirectory.framework/Frameworks/OpenDirectory.framework"),

File "/System/Library/Frameworks/Python.framework/Versions/2.7/Extras/lib/python/PyObjC/objc/_dylld.py", line 122, in pathForFramework

fpath, name, version = infoForFramework(dylld_find(path))

File "/System/Library/Frameworks/Python.framework/Versions/2.7/Extras/lib/python/PyObjC/objc/_dylld.py", line 117, in dylld_find

return dylld_framework(filename, framework_name, version)

File "/System/Library/Frameworks/Python.framework/Versions/2.7/Extras/lib/python/PyObjC/objc/_dylld.py", line 88, in dylld_framework

raise ImportError("Framework %s could not be found" % (framework_name,))

ImportError: Framework OpenDirectory could not be found

AVFoundation
Message
OpenDirectory
PubSub

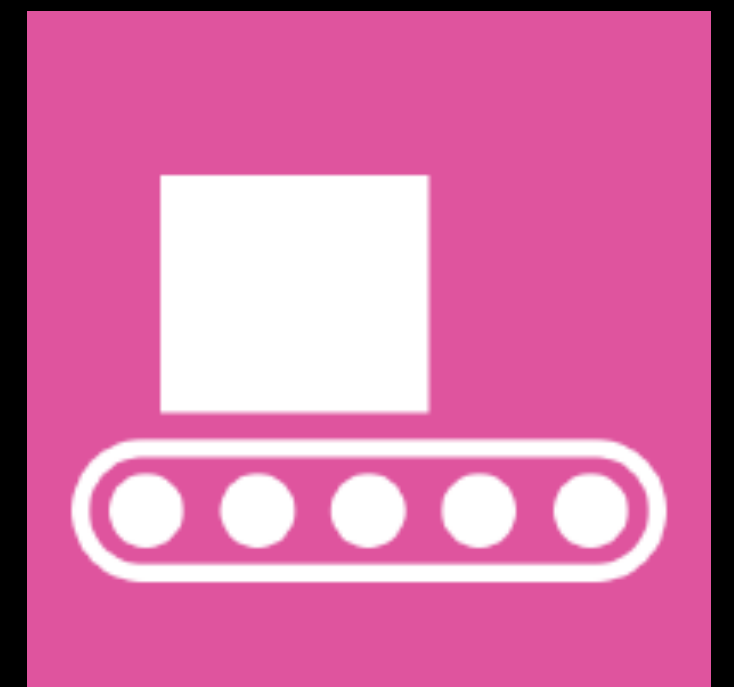
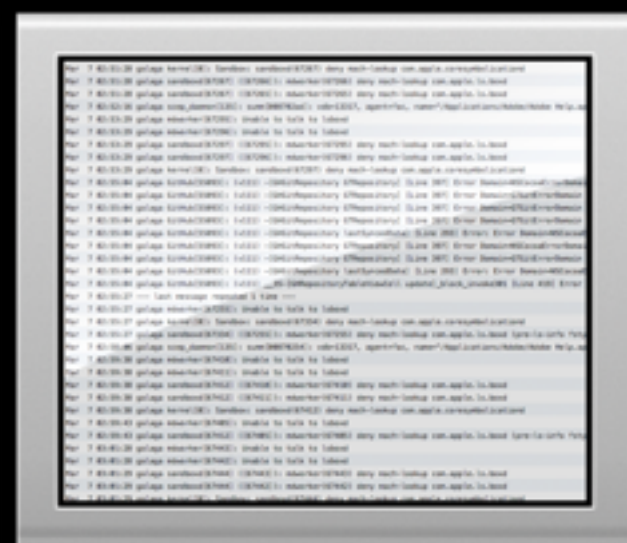
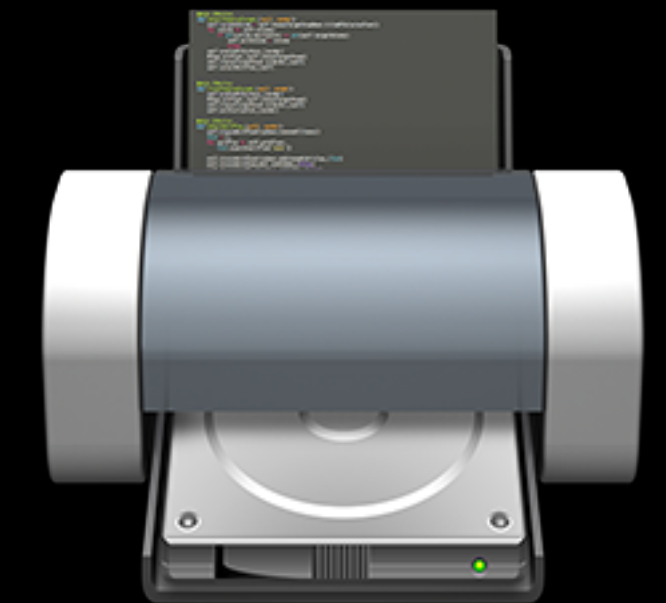
Python 2 on macOS

- Apple is warning us it's going away in a future macOS release
- It's still there in Big Sur Beta 6
- But: **it's partially broken!**



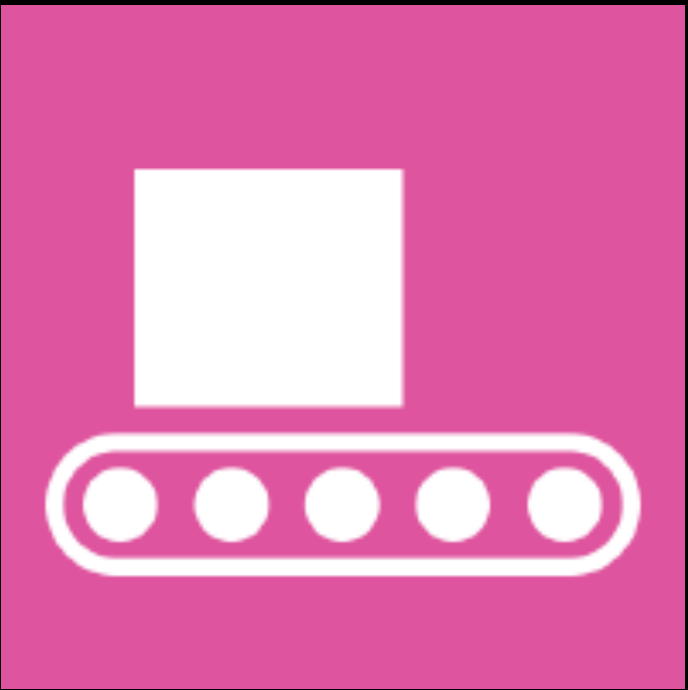
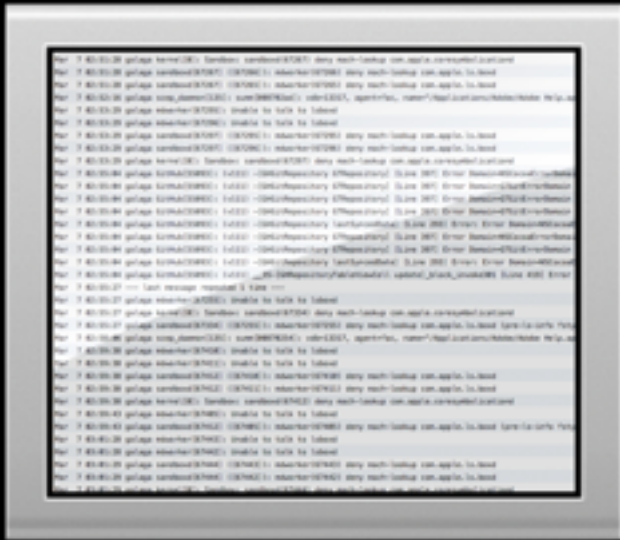
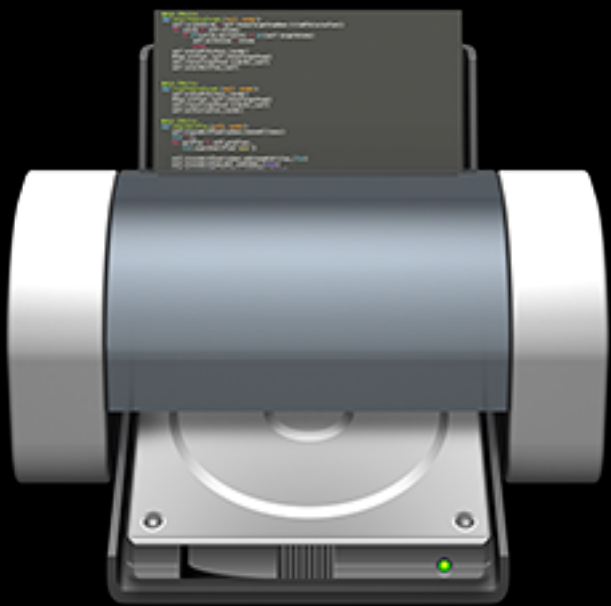
Stuff is gonna break.



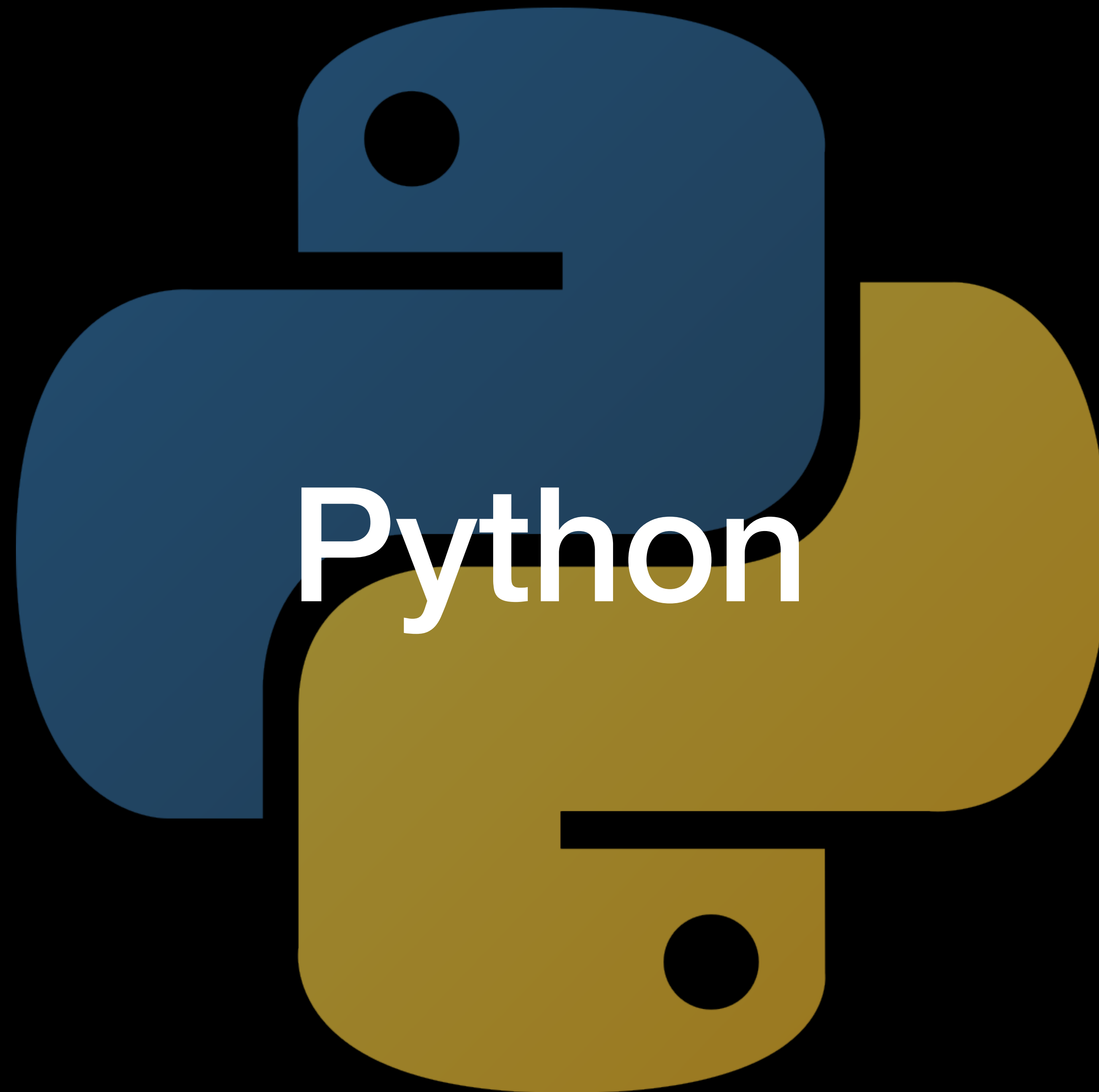


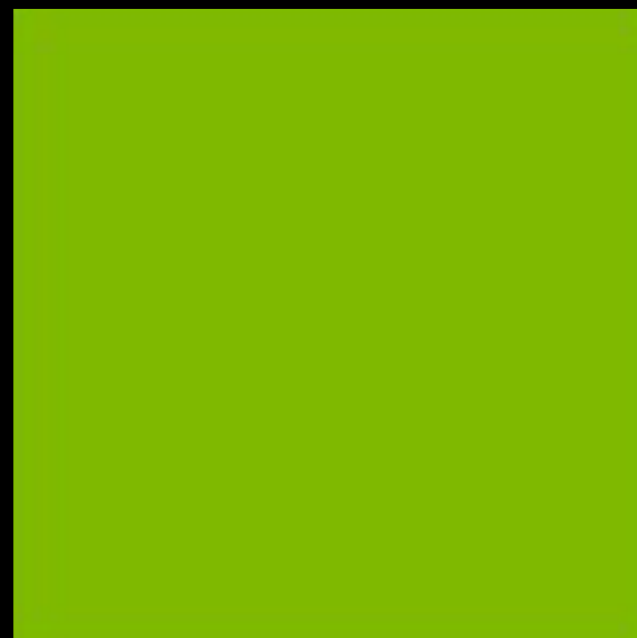
AutoDMG
AutoPkg
Crypt-Server
dockutil
Imagr
installapplications
installinstallmacos.py
Loginl oq

pycreateuserpkg
pyfacts
Recipe Robot
reposado
Sal
umad
vfuse
Zentral











< Back

Install Scripts

2019

- ▼ Installer Package
 - Distribution
- ▼ Microsoft_Word.pkg
 - preinstall
 - postinstall
 - chain_app
- ▼ Office16_all_autoupdate.pkg
 - preinstall
 - postinstall
- ▼ Office16_all_licensing.pkg
 - postinstall
 - dockutil

dockutil Show in Suspicious Package

Kind Python script
Where MSWord2019-16.29.19091700.pkg/Office16_all_licensing.pkg/Scripts/dockutil
When Unknown, but likely referenced by another script

```
#!/usr/bin/python

# This file is based on or incorporates material from the projects listed below (Third Party IP
# which Microsoft received such Third Party IP, are set forth below. Such licenses and notices
# licenses the Third Party IP to you under the licensing terms for the Microsoft product. Micro
# under this agreement, whether by implication, estoppel or otherwise.

# DOCKUTIL
# Copyright 2008 Kyle Crawford

# Provided for Informational Purposes Only

# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0

# THIS CODE IS PROVIDED ON AN *AS IS* BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITH
# IMPLIED WARRANTIES OR CONDITIONS OF TITLE, FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY C

# See the Apache Version 2.0 License for specific language governing permissions and limitation
```



MSWord2019-16.29.19091700.pkg



Back

Install Scripts

- ▼ Installer Package
 - Distribution
- ▼ Microsoft_Word.pkg
 - preinstall
 - postinstall
 - chain_app
- ▼ Office16_all_autoupdate.pkg
 - preinstall
 - postinstall
- ▼ Office16_all_licensing.pkg
 - postinstall
 - dockutil**

dockutil

[Show in Suspicious Package](#)

Kind Python script

Where MSWord2019-16.29.19091700.pkg/Office16_all_licensing.pkg/Scripts/dockutil

When Unknown, but likely referenced by another script

2019

`#!/usr/bin/python`

```
# This file is based on or incorporates material from the projects listed below (Third Party IP
# which Microsoft received such Third Party IP, are set forth below. Such licenses and notices
# licenses the Third Party IP to you under the licensing terms for the Microsoft product. Micro
# under this agreement, whether by implication, estoppel or otherwise.
```

```
# DOCKUTIL
# Copyright 2008 Kyle Crawford
```

```
# Provided for Informational Purposes Only
```

```
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
```

```
# http://www.apache.org/licenses/LICENSE-2.0
```

```
# THIS CODE IS PROVIDED ON AN *AS IS* BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITH
# IMPLIED WARRANTIES OR CONDITIONS OF TITLE, FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY C
```

```
# See the Apache Version 2.0 License for specific language governing permissions and limitation
```




MSWord2019-16.40.20081000.pkg



Back

Install Scripts

2020

- ▼ Installer Package
 - Distribution
- ▼ Microsoft_Word.pkg
 - preinstall**
 - postinstall

preinstall

Show in Suspicious Package

Kind Z Shell script

Where MSWord2019-16.40.20081000.pkg/Microsoft_Word.pkg/Scripts/preinstall

When Before moving files into place

```
#!/bin/zsh
```

```
APPNAME= Microsoft Word"
```

```
LOGDIRECTORY="/Library/Logs/Microsoft"
```

```
LOGFILE="$LOGDIRECTORY/preinstall.log"
```

```
# Ensure we have a log directory before logging  
/bin/mkdir -p "$LOGDIRECTORY"
```

```
# Logs a string of text to the default log file  
function write_log()  
{
```

```
    message="$1"
```

```
    timestamp=$(/bin/date +%Y-%m-%d' '%T)
```

```
    echo "$timestamp [Apple Installer - preinstall] <Info> $message" >> $LOGFILE
```

```
}
```

```
# Tests whether the app to be updated is closed - returns 1 if closed, or 0 if running  
function is_app_closed()  
{
```

```
    /usr/bin/pgrep -q "$1"
```

```
}
```





Back



Path



Action



Get Info



Installer

Q Search

Package Info

All Files

- postinstall
- Tools
 - 7za
 - AcroPatchInstall....
 - Contents
 - beta.ARMOverrid...
 - CheckSpace.py
 - com.adobe.acrob...
 - com.adobe.acrob...
 - FeatureLockDow...
 - FeatureLockDow...
 - mergeplists.py
 - PatchRequisite.pl...
 - PDFAdminSetting...
 - TaskExecutor.py**
 - Tasks.plist
 - trunk.ARMOverri...

aam.pkg

- preinstall
- postinstall
- CheckForIgnoreAAM
- Install.app
- packages

```
1  #!/usr/bin/python
2
3  import sys,os,plistlib,pwd,grp,stat,urllib,subprocess,traceback
4
5  taskPlist = unicode(sys.argv[2], sys.getfilesystemencoding())
6  appPath = unicode(sys.argv[3], sys.getfilesystemencoding())
7  patchStore = unicode(sys.argv[4], sys.getfilesystemencoding())
8  binaryPath = unicode(sys.argv[5], sys.getfilesystemencoding())
9  CLI_mode = sys.argv[6]
10 title = sys.argv[7]
11
12 rollbackStore = ''
13 if len(sys.argv) == 9:
14     rollbackStore = unicode(sys.argv[8], sys.getfilesystemencoding())
15
16 rollbackTask = []
17
18 errorCodeDict = {
19     "PatchTaskApplyFailed":107,
20     "PathAbsentForAttributesChange":108,
21     "ChangeOwnershipFailed":109,
22     "ChangePermissionFailed":110,
23     "OverwriteFileFailed":111,
24     "RemovePathFailed":112,
25     "FailedAttributesCheck":113,
26     "FailedChecksumMatch":114,
27     "RollbackError":115,
28     "GenericError":116,
29 }
30
```

2019



Back



Path



Action



Get Info



Installer

Q Search

Package Info

All Files

- postinstall
- Tools
 - 7za
 - AcroPatchInstall....
 - Contents
 - beta.ARMOverrid...
 - CheckSpace.py
 - com.adobe.acrob...
 - com.adobe.acrob...
 - FeatureLockDow...
 - FeatureLockDow...
 - mergeplists.py
 - PatchRequisite.pl...
 - PDFAdminSetting...
 - TaskExecutor.py
 - Tasks.plist
 - trunk.ARMOverri...

aam.pkg

- preinstall
- postinstall
- CheckForIgnoreAAM
- Install.app
- packages

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

```
import sys,os,plistlib,pwd,grp,stat,urllib,subprocess,traceback

taskPlist = unicode(sys.argv[2], sys.getfilesystemencoding())
appPath = unicode(sys.argv[3], sys.getfilesystemencoding())
patchStore = unicode(sys.argv[4], sys.getfilesystemencoding())
binaryPath = unicode(sys.argv[5], sys.getfilesystemencoding())
CLI_mode = sys.argv[6]
title = sys.argv[7]

rollbackStore = ''
if len(sys.argv) == 9:
    rollbackStore = unicode(sys.argv[8], sys.getfilesystemencoding())

rollbackTask = []

errorCodeDict = {
    "PatchTaskApplyFailed":107,
    "PathAbsentForAttributesChange":108,
    "ChangeOwnershipFailed":109,
    "ChangePermissionFailed":110,
    "OverwriteFileFailed":111,
    "RemovePathFailed":112,
    "FailedAttributesCheck":113,
    "FailedChecksumMatch":114,
    "RollbackError":115,
    "GenericError":116,
}
```

2019

2020



Back



Path



Action



Get Info



Installer

Package Info

All Files

Installer Package

Distribution

payload.pkg

preinstall

postinstall

▼ Tools

7za

AcroPatchInstall....

beta.ARMOverrid...

CheckSpace.py

com.adobe.acrob...

com.adobe.acrob...

FeatureLockDow...

FeatureLockDow...

mergeplists.py

PatchRequisite.pl...

PDFAdminSetting...

TaskExecutor.py

Tasks.plist

trunk.ARMOverri...

AcroServicesUpdater.pkg

postinstall

```
1  #!/usr/bin/python
2  ##to check available space on startup volume
3
4  import sys,os,statvfs,subprocess
5
6  spaceErrorCode = 107
7  genericErrorCode = 116
8
9  def runCmd(cmd):
10     status = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
11     (output,error) = status.communicate()
12     returncode = status.returncode
13     if output != '':
14         print ("%s" % (output))
15
16     return (output,error,returncode)
17
18  def checkSpace(plistFile):
19     retval = 0
20
21     if os.path.isfile(plistFile):
22         plist = plistFile.rstrip(".plist")
23         cmd = '/usr/bin/defaults read ' + "'" + plist + "'" + ' SpaceBytes'
24         (output,error,returncode) = runCmd(cmd)
25         if returncode != 0:
26             print ("ERROR: Encountered error in reading PatchRequisite.plist.")
27             if error != '':
28                 print error
29             return(genericErrorCode, 0, 0)
30
31     output = output.rstrip("\n")
```

2020



Back

Install Scripts

▼ Installer Package

■ Distribution

▼ payload.pkg

■ preinstall

■ postinstall

▼ Tools

■ 7za

▶ AcroPatchInstall.app

■ beta.ARMOOverrides.plist

■ CheckSpace.py

■ com.adobe.acrobat.dc.plist

■ com.adobe.acrobat.servicesupdater....

■ FeatureLockDown_Old

■ FeatureLockDown_Updated

■ mergeplists.py

■ PatchRequisite.plist

■ PDFAdminSettings.plist

■ TaskExecutor.py

■ Tasks.plist

■ trunk.ARMOOverrides.plist

▼ AcroServicesUpdater.pkg

■ postinstall

preinstall

Show in Suspicious Package

Kind Bourne-Again Shell script

Where AcrobatDCUpd2001220043.pkg/payload.pkg/Scripts/preinstall

When Before moving files into place

#!/bin/bash

#####

#This will be auto updated during the creation of incremental patch installer in mkpackages
isPatchIncr=False

#####

function exitWithErrorCode
{

rm -rf \$installation_path

echo "ERROR: Preinstall exited with error code \$1."

if [\$targetApptxtPresent -eq 1]

then

echo \$1 > "\$errorCodeFile"

fi

exit \$1

}

function checkDefaultPathAndObtainFromOSIfInvalid

{

echo "Info: Checking the default path \$defaultPath."

if [! -e "\$defaultPath"]

then

echo "The default path \$defaultPath does not exist. Using OS calls to find
\$acropatchinstall ObtainPathFromOS com.adobe.Acrobat.Pro \$installation_path

else

\$acropatchinstall ObtainPathFromOS com.adobe.Acrobat.Pro \$installation_path



[https://github.com/
scriptingosx/pkggcheck](https://github.com/scriptingosx/pkggcheck)


```
bash-3.2$ ./pkgcheck.sh /Volumes/AcrobatDCUpd2001220043/AcrobatDCUpd2001220043.pkg
AcrobatDCUpd2001220043
/Volumes/AcrobatDCUpd2001220043/AcrobatDCUpd2001220043.pkg
Signature:      Developer ID Installer: Adobe Systems, Inc. (JQ525L2MZD)
Notarized:      Yes, Notarized Developer ID
Type:           Flat Distribution PKG
Contains 2 component pkgs
```

```

Type:           Flat Component PKG
Identifier:      com.adobe.AcroServicesUpdater
Version:         0
Location:        /tmp/webinstaller
Contains 1 resource files
```

```

Type:           Flat Component PKG
Identifier:      com.adobe.acrobat.AcrobatDCUpd2001220043
Version:         20.012.20043
Location:        /tmp/com.adobe.acrobat.updater
Contains 94 resource files
postinstall has shebang #!/bin/bash
Tools/CheckSpace.py has shebang #!/usr/bin/python
Tools/mergeplists.py has shebang #!/usr/bin/python
Tools/TaskExecutor.py has shebang #!/usr/bin/python
preinstall has shebang #!/bin/bash
```

```
bash-3.2$ ./pkgcheck.sh /Volumes/AcrobatDCUpd2001220043/AcrobatDCUpd2001220043.pkg
AcrobatDCUpd2001220043
/Volumes/AcrobatDCUpd2001220043/AcrobatDCUpd2001220043.pkg
Signature:      Developer ID Installer: Adobe Systems, Inc. (JQ525L2MZD)
Notarized:      Yes, Notarized Developer ID
Type:           Flat Distribution PKG
Contains 2 component pkgs
```

```
Type:           Flat Component PKG
Identifier:      com.adobe.AcroServicesUpdater
Version:        0
Location:       /tmp/webinstaller
Contains 1 resource files
```

```
Type:           Flat Component PKG
Identifier:      com.adobe.acrobat.AcrobatDCUpd2001220043
Version:        20.012.20043
Location:       /tmp/com.adobe.acrobat.updater
Contains 94 resource files
```

```
postinstall has shebang #!/bin/bash
Tools/CheckSpace.py has shebang #!/usr/bin/python
Tools/mergeplists.py has shebang #!/usr/bin/python
Tools/TaskExecutor.py has shebang #!/usr/bin/python
preinstall has shebang #!/bin/bash
```


- 
- The Python logo, consisting of two interlocking snakes, one blue and one yellow, is centered in the background. The blue snake is on the left and the yellow snake is on the right, both facing each other.
- Third-party tools
 - Third-party package scripts
 - Your own scripts (and pkgs)

Replacing missing Python

[https://www.python.org/
downloads/mac-osx/](https://www.python.org/downloads/mac-osx/)



Donate

Search GO

Python >>> Downloads >>> Mac OS X

Python Releases for Mac OS X

- [Latest Python 3 Release - Python 3.8.5](#)
- [Latest Python 2 Release - Python 2.7.18](#)

Stable Releases

- [Python 3.7.9 - Aug. 17, 2020](#)
 - Download [macOS 64-bit installer](#)

Pre-releases

- [Python 3.5.10rc1 - Aug. 22, 2020](#)
 - No files for this release.

Python.org framework issues

- Install to /Library/Frameworks/Python.framework
(can't easily have Python 2 and 3 installed at the same time)
- Break if you move/rename
- Don't include PyObjC, xattr, or other "Apple" bits

[https://github.com/gregneagle/
relocatable-python](https://github.com/gregneagle/relocatable-python)

<

>

github.com/gregneagle/relocatable-python/

Search or jump to...

/

Pull requests

Issues

Marketplace

Explore

gregneagle / relocatable-python

Unwatch

9

Star

80

Fork

9

<> Code

Issues 1

Pull requests 1

Actions

Projects

Wiki

Security

Insights

Settings

main

1 branch

0 tags

Go to file

Add file

Code

gregneagle

Python 3 fix for localizing shebangs in Python.framework/Versions/Cur...

8bce58e on Feb 8

18 commits

locallibs	Python 3 fix for localizing shebangs in Python.framework/Versions/Cu...	7 months ago
.gitignore	First code commit	2 years ago
LICENSE	Initial commit	2 years ago
README.txt	Update README.txt	9 months ago
make_relocatable_python_framework...	Make scripts in Python.framework/Versions/Current/bin relocatable. F...	7 months ago
research_notes.txt	First code commit	2 years ago

README.txt

This is a tool to make a relocatable Python framework containing PyObjC.

NOTE: while the resulting frameworks (and interpreters) have been successfully used in several projects (among them Imagr, Munki, and AutoPkg) there is no guarantee it is suitable as a general-purpose Python installation.

A relocatable Python.framework is ideal for embedding into an application's Frameworks directory, and can even be

About

A tool for building standalone relocatable Python.framework bundles

Readme

Apache-2.0 License

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 2

```
$ ./make_relocatable_python_framework.py --help
Usage: make_relocatable_python_framework.py [options]
```

Options:

- h, --help show this help message and exit
- destination=DESTINATION
 Directory destination for the Python.framework
- baseurl=BASEURL Override the base URL used to download the framework.
- os-version=OS_VERSION
 Override the macOS version of the downloaded pkg.
 Current supported versions are "10.6" and "10.9". Not
 all Python version and macOS version combinations are
 valid.
- python-version=PYTHON_VERSION
 Override the version of the Python framework to be
 downloaded. See available versions at
 <https://www.python.org/downloads/mac-osx/>
- pip-requirements=PIP_REQUIREMENTS
 Path to a pip freeze requirements.txt file that
 describes extra Python modules to be installed. If not
 provided, certain useful modules for macOS will be
 installed.


```
$ ./make_relocatable_python_framework.py --python-version=3.8.5 \  
                                         --destination ~/Desktop/Python3.framework
```

```
$ ./make_relocatable_python_framework.py --python-version=3.8.5 \
--destination ~/Desktop/Python3.framework
Downloading https://www.python.org/ftp/python/3.8.5/python-3.8.5-macosx10.9.pkg...
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 28.8M  100 28.8M    0     0   910k      0  0:00:32  0:00:32 --:--:--  705k
Expanding /var/folders/tc/sd4_mtvj14jdy7cg21m2gmcw000495/T/tmpKhCNA...
Extracting /var/folders/tc/sd4_mtvj14jdy7cg21m2gmcw000495/T/tmpKhCNA__expanded__/_
Python_Framework.pkg/Payload to /Users/gneagle/Desktop/Python3.framework/Python.framework...
Ensuring correct modes for files in /Users/gneagle/Desktop/Python3.framework/Python.framework...
Analyzing /Users/gneagle/Desktop/Python3.framework/Python.framework...
.....
/usr/bin/install_name_tool -id @rpath/Versions/3.8/Python /Users/gneagle/Desktop/
Python3.framework/Python.framework/Versions/3.8/Python
/usr/bin/install_name_tool -change /Library/Frameworks/Python.framework/Versions/3.8/Python
@rpath/Versions/3.8/Python /Users/gneagle/Desktop/Python3.framework/Python.framework/Versions/
3.8/bin/python3.8
/usr/bin/install_name_tool -change /Library/Frameworks/Python.framework/Versions/3.8/Python
@rpath/Versions/3.8/Python /Users/gneagle/Desktop/Python3.framework/Python.framework/Versions/
3.8/Resources/Python.app/Contents/MacOS/Python

/usr/bin/install_name_tool -id @rpath/Versions/3.8/lib/libtcl8.6.dylib /Users/gneagle/Desktop/
Python3.framework/Python.framework/Versions/3.8/lib/libtcl8.6.dylib
```

Using cached pyobjc_framework_Automator-6.2.2-py2.py3-none-any.whl (4.9 kB)
Collecting pyobjc-framework-AppleScriptObjC==6.2.2; platform_release >= "10.0"
Using cached pyobjc_framework_AppleScriptObjC-6.2.2-py2.py3-none-any.whl (3.9 kB)
Collecting pyobjc-framework-EventKit==6.2.2; platform_release >= "12.0"
Using cached pyobjc_framework_EventKit-6.2.2-py2.py3-none-any.whl (5.6 kB)
Collecting pyobjc-framework-InputMethodKit==6.2.2; platform_release >= "9.0"
Using cached pyobjc_framework_InputMethodKit-6.2.2-cp36-abi3-macosx_10_9_x86_64.whl (7.5 kB)
Collecting pyobjc-framework-LaunchServices==6.2.2
Using cached pyobjc_framework_LaunchServices-6.2.2-py2.py3-none-any.whl (3.3 kB)
Collecting pyobjc-framework-ModelIO==6.2.2; platform_release >= "15.0"
Using cached pyobjc_framework_ModelIO-6.2.2-cp36-abi3-macosx_10_9_x86_64.whl (13 kB)
Collecting pyobjc-framework-CoreMedia==6.2.2; platform_release >= "11.0"
Using cached pyobjc_framework_CoreMedia-6.2.2-cp36-abi3-macosx_10_9_x86_64.whl (20 kB)
Collecting pyobjc-framework-VideoSubscriberAccount==6.2.2; platform_release >= "18.0"
Using cached pyobjc_framework_VideoSubscriberAccount-6.2.2-py2.py3-none-any.whl (3.6 kB)
Collecting pyobjc-framework-Metal==6.2.2; platform_release >= "15.0"
Using cached pyobjc_framework_Metal-6.2.2-cp36-abi3-macosx_10_9_x86_64.whl (21 kB)
Collecting pyobjc-framework-CoreText==6.2.2
Using cached pyobjc_framework_CoreText-6.2.2-cp36-abi3-macosx_10_9_x86_64.whl (25 kB)
Collecting pyobjc-framework-CoreLocation==6.2.2; platform_release >= "10.0"
Using cached pyobjc_framework_CoreLocation-6.2.2-cp36-abi3-macosx_10_9_x86_64.whl (9.0 kB)
Collecting pyobjc-framework-OpenDirectory==6.2.2; platform_release >= "10.0"
Using cached pyobjc_framework_OpenDirectory-6.2.2-py2.py3-none-any.whl (12 kB)
Collecting pyobjc-framework-CloudKit==6.2.2; platform_release >= "14.0"
Using cached pyobjc_framework_CloudKit-6.2.2-py2.py3-none-any.whl (7.1 kB)

bin/pip3...

Modifying shebang for /Users/gneagle/Desktop/Python3.framework/Python.framework/Versions/3.8/
bin/easy_install-3.8...

Modifying shebang for /Users/gneagle/Desktop/Python3.framework/Python.framework/Versions/3.8/
bin/pydoc3.8...

Modifying shebang for /Users/gneagle/Desktop/Python3.framework/Python.framework/Versions/3.8/
bin/idle3.8...

Modifying shebang for /Users/gneagle/Desktop/Python3.framework/Python.framework/Versions/3.8/
bin/python3.8-config...

Modifying shebang for /Users/gneagle/Desktop/Python3.framework/Python.framework/Versions/3.8/
bin/2to3-3.8...

Modifying shebang for /Users/gneagle/Desktop/Python3.framework/Python.framework/Versions/3.8/
bin/xattr...

Done!

Customized, relocatable framework is at /Users/gneagle/Desktop/Python3.framework/
Python.framework

bash-3.2\$



Python3.framework

Python.framework/Versions/3.8/bin/python3

```
$ sudo ln -s /Users/Shared/Python3.framework/Versions/3.8/bin/python3 /usr/local/bin/python3
```


/Library/MyOrg/Python.framework
/Library/MyOrg/Python2.framework
/Library/MyOrg/Python3.framework

[https://github.com/
macadmins/python](https://github.com/macadmins/python)

📁 macadmins / python

👁 Watch ▾

13

★ Star

40

🍴 Fork

4

A Python 3 framework that currently installs to `/Library/ManagedFrameworks/Python/Python3.framework` .

Please see Apple's documentation on [file system basics](#) for more information on the thought process here.

This is an intended replacement for when Apple removes `/usr/bin/python`

Using interactively

After installing any of the packages, a symbolic link can be used within terminal for interactive Python sessions. At the time of this writing `/usr/local/bin/managed_python3` points to

`/Library/ManagedFrameworks/Python/Python3.framework/Versions/Current/bin/python3`

Using with scripts

It is currently recommended to point directly to symbolic link provided by the Python framework.

At the time of this writing

`/Library/ManagedFrameworks/Python/Python3.framework/Versions/Current/bin/python3` points to

`/Library/ManagedFrameworks/Python/Python3.framework/Versions/3.8/bin/python3.8`

An example script would look like the following:

```
#!/Library/ManagedFrameworks/Python/Python3.framework/Versions/Current/bin/python3

print('This is an example script.')
```


But this doesn't help
with...

[< Back](#)

Install Scripts

▼ **Installer Package**

■ Distribution

▼ **payload.pkg**

■ preinstall

■ postinstall

▼ **Tools**

■ 7za

▶ AcroPatchInstall.app

■ beta.ARMOOverrides.plist

■ **CheckSpace.py**

■ com.adobe.acrobat.dc.plist

■ com.adobe.acrobat.servicesupdater....

■ FeatureLockDown_Old

■ FeatureLockDown_Updated

■ mergeplists.py

■ PatchRequisite.plist

■ PDFAdminSettings.plist

■ TaskExecutor.py

■ Tasks.plist

■ trunk.ARMOOverrides.plist

▼ **AcroServicesUpdater.pkg**

■ postinstall

CheckSpace.py

[➔ Show in Suspicious Package](#)

Kind Python script

Where AcrobatDCUpd2001220043.pkg/payload.pkg/Scripts/Tools/CheckSpace.py

When Unknown, but likely referenced by another script

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

```
## To check available space on startup volume
```

```
import sys,os,statvfs,subprocess
```

```
spaceErrorCode = 107
```

```
genericErrorCode = 116
```

```
def runCmd(cmd):
```

```
    status = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE, stderr=su
```

```
    (output,error) = status.communicate()
```

```
    returncode = status.returncode
```

```
    if output != '':
```

```
        print ("%s") %(output)
```

```
    return (output,error,returncode)
```

```
def checkSpace(plistFile):
```

```
    retval = 0
```

```
    if os.path.isfile(plistFile):
```

```
        plist = plistFile.rstrip(".plist")
```

```
        cmd = '/usr/bin/defaults read ' + "'" + plist + "'" + ' SpaceBytes'
```

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


`/usr/bin/python`

So why is python 2 still
available in Big Sur?

xattr

```
xattr -d com.apple.quarantine some_file_from_the_internet
```



```
bash-3.2$ sw_vers
```

```
ProductName:  macOS
```

```
ProductVersion: 11.0
```

```
BuildVersion: 20A5364e
```

```
bash-3.2$ file /usr/bin/xattr
```

```
/usr/bin/xattr: Python script text executable, ASCII text
```

```
bash-3.2$ head -1 /usr/bin/xattr
```

```
#!/System/Library/Frameworks/Python.framework/Versions/2.7/Resources/Python.app/Contents/MacOS/Python
```


MacSysAdmin 2021

– European Macintosh System Administrators Meeting 2021 –



Some choices

- Stick with **Python 2**
(and ship/install your own Python 2 framework)
- Move to **Python 3**
(and ship/install your own Python 3 framework)
- Move to **Objective-C/Swift/Go...**
- Convert simple scripts to **/bin/sh** or **/bin/zsh**

- Stick with Python 2
(and ship/install your own Python 2 framework)

Python 2 EOL Jan 2020
= no more security patches

- Convert simple scripts to `/bin/sh` or `/bin/zsh`
(not bash!)

- Move to Objective-C/Swift/Go...



Search or jump to...

github.com/gregneagle/pycreateuserpkg



 gregneagle / pycreateuserpkg

Unwatch ▼

28

★ Star

204

 Fork

31

Code

Issues 10

Pull requests 5

Projects 0

 Wiki Security

Insights

 Settings

Python tool to create macOS packages that create user accounts

Edit

Manage topics

64 commits

1 branch

 **0** releases

 5 contributors

 [View license](#)

Branch: master ▼

New pull request

Create new file

Upload files

Find file

Clone or download ▼

 gregneagle Fix busted merge of authentication_authority branch	Latest commit c3f04a3 yesterday
 createuser	Fix busted merge of authentication_authority branch yesterday
 locallibs	Replace create_user.py tool with Objective-C createuser tool; elimina... 2 months ago
 pkg_scripts	Fix busted merge of authentication_authority branch yesterday
 .gitignore	Add Objective-C version of createuser tool 2 months ago
 LICENSE.md	More licensing info 2 years ago
 README.md	When updating an existing account on the current boot disk, don't cha... 2 days ago
 createuserpkg	added in user password hint (#33) 5 months ago

 README.md

Tool for generating packages that create macOS user accounts on 10.8-10.15

user.pkg postinstall

```
#!/bin/bash
#
# postinstall for local account install
#
# PlistArrayAdd() by Per Olofsson (aka MagerValp); parts of the rest of the
# script also indebted to work by Per

PlistArrayAdd() {
    # Add $value to $array_name in $plist_path, creating if necessary
    local plist_path="$1"
    local array_name="$2"
    local value="$3"
    local old_values
    local item

    old_values=$(/usr/libexec/PlistBuddy -c "Print :$array_name" "$plist_path" 2>/dev/null)
    if [[ $? == 1 ]]; then
        # Array doesn't exist, create it
        /usr/libexec/PlistBuddy -c "Add :$array_name array" "$plist_path"
    else
```



```
fi
# Add item to array
/usr/libexec/PlistBuddy -c "Add array item: string '$value'" "$plist_path"
}
```

user.pkg postinstall

```
# set our current directory (the one this script is in)
# dirname is not available in Recovery, so we use this cryptic shell-ism
MYDIR=${0%/*}
```

```
# find and source our config file to set our variables
source "$MYDIR/config"
```

```
if [ "$3" == "/" ]; then
    # create user using OpenDirectory APIs
    "$MYDIR"/create_user.py "$MYDIR/$USERNAME.plist"
    # we're operating on the boot volume
    if [ "$USER_IS_ADMIN" == "True" ]; then
        # add $USERNAME to admin group
        /usr/sbin/dseditgroup -o edit -a "$USERNAME" -t user admin
    else
        # remove $USERNAME from admin group
        /usr/sbin/dseditgroup -o edit -d "$USERNAME" -t user admin
    fi
    if [ "$ENABLE_AUTOLOGIN" == "True" ]; then
        # conv kcpassword into place
```

```
fi
# Add item to array
/usr/libexec/PlistBuddy -c "Add $array_name:strings N'$package'" "$plist_path"
}
```

```
# set our current directory (the one this script is in)
# dirname is not available in Recovery, so we use this cryptic shell-ism
MYDIR=${0%/*}
```

```
# find and source our config file to set our variables
source "$MYDIR/config"
```

```
if Γ "$3" == "/" then
```

```
# create user using OpenDirectory APIs
"$MYDIR"/create_user.py "$MYDIR/$USERNAME.plist"
```

```
# we're operating on the boot volume
```

```
if [ "$USER_IS_ADMIN" == "True" ]; then
```

```
# add $USERNAME to admin group
```

```
/usr/sbin/dseditgroup -o edit -a "$USERNAME" -t user admin
```

else

```
# remove $USERNAME from admin group
```

```
/usr/sbin/dseditgroup -o edit -d "$USERNAME" -t user admin
```

fi

```
if [ "$ENABLE_AUTOLOGIN" == "True" ]; then
```

```
# copy keypassword into place
```

```
create_user.py

1  #!/usr/bin/python
2
3  # Copyright 2018 Greg Neagle.
4  #
5  # Licensed under the Apache License, Version 2.0 (the "License");
6  # you may not use this file except in compliance with the License.
7  # You may obtain a copy of the License at
8  #
9  #     http://www.apache.org/licenses/LICENSE-2.0
10 #
11 # Unless required by applicable law or agreed to in writing, software
12 # distributed under the License is distributed on an "AS IS" BASIS,
13 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 # See the License for the specific language governing permissions and
15 # limitations under the License.
16
17 """Creates a local user from a user.plist using OpenDirectory API"""
18
19 import os
20 import sys
21
22 # PyLint cannot properly find names inside Cocoa libraries, so issues bogus
23 # No name 'Foo' in module 'Bar' warnings. Disable them.
24 # pylint: disable=E0611
25 from Foundation import NSData
26 from Foundation import NSPropertyListSerialization
27 from Foundation import NSPropertyListMutableContainers
28 from OpenDirectory import ODSession, ODNode, kODRecordTypeUsers
29 # pylint: enable=E0611
30
```



```
16
17 """Creates a local user from a user.plist using OpenDirectory API"""
18
19 import os
20 import sys
21
22 # PyLint cannot properly find names inside Cocoa libraries, so issues bogus
23 # No name 'Foo' in module 'Bar' warnings. Disable them.
24 # pylint: disable=E0611
25 from Foundation import NSData
26 from Foundation import NSPropertyListSerialization
27 from Foundation import NSPropertyListMutableContainers
28 from OpenDirectory import ODSession, ODNNode, kODRecordTypeUsers
29 # pylint: enable=E0611
30
31
32 def major_darwin_version():
33     '''Returns an integer with the current major Darwin version'''
34     return int(os.uname()[2].split('.')[0])
35
36
37 def local_node():
38     """Returns local DS node"""
39     my_session = ODSession.defaultSession()
40     if not my_session:
41         return None
42     node, err = ODNNode.nodeWithSession_name_error_(
43         my_session, "/Local/Default", None
44     )
45     if err:
46         print >> sys.stderr, err
47     return node
48
```

createuser > My Mac

createuser | Build createuser: **Succeeded** | Today at 2:17 PM

createuser > createuser > main.m > mergeAuthenticationAuthorities()

createuser

createuser

main.m

Products

createuser

1 //

2 // main.m

3 // createuser

4 //

5 // Created by Greg Neagle on 8/13/19.

6 //

7 // Copyright 2019 Greg Neagle.

8 //

9 // Licensed under the Apache License, Version 2.0 (the "License");

10 // you may not use this file except in compliance with the License.

11 // You may obtain a copy of the License at

12 //

13 // <http://www.apache.org/licenses/LICENSE-2.0>

14 //

15 // Unless required by applicable law or agreed to in writing, software

16 // distributed under the License is distributed on an "AS IS" BASIS,

17 // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

18 // See the License for the specific language governing permissions and

19 // limitations under the License.

20

21

22 #include <libgen.h> // for basename()

23

24 #import <Foundation/Foundation.h>

25 #import <OpenDirectory/OpenDirectory.h>

26

27 NSString * kAuthenticationAuthorityKey = @"authentication_authority";

28

29 ODSNode * localDSNode() {

30 // returns local DS node

31 ODSession * mySession = [ODSession defaultSession];

32 if (mySession == nil) {

33 return nil;

34 }

35 NSError * err = nil;

36 ODSNode *node = [ODNode nodeWithSession:mySession name:@"/Local/Default" error: &err];

37 if (err != nil) {

38 NSLog(@"Could not get local DS node: %@", err);

39 }

40 return node;

Identity and Type

Name main.m

Type Default - Objective-C Sou...

Location Relative to Group

main.m

Full Path /Users/Shared/code/pycreateuserpkg/createuser/createuser/main.m

On Demand Resource Tags

Only resources are taggable

Target Membership

☒ createuser

Text Settings

Text Encoding No Explicit Encoding

Line Endings No Explicit Line Endings

Indent Using Spaces

Widths 4 4

Tab Indent

☒ Wrap lines

postinstall

```
fi
# Add item to array
/usr/libexec/PlistBuddy -c "Add :$array_name string \"$value\"" "$plist_path"
}
```

```
# set our current directory (the one this script is in)
# dirname is not available in Recovery, so we use this cryptic shell-ism
MYDIR=${0%/*}
```

```
# find and source our config file to set our variables
source "$MYDIR/config"
```

```
if [ "$3" == "/" ]; then
```

```
# create user using OpenDirectory APIs
"$MYDIR"/create_user.py "$MYDIR/$USERNAME.plist"
```

```
# we're operating on the boot volume
```

```
if [ "$USER_IS_ADMIN" == "True" ]; then
```

```
# add $USERNAME to admin group
```

```
/usr/sbin/dseditgroup -o edit -a "$USERNAME" -t user admin
```

```
else
```

```
# remove $USERNAME from admin group
```

```
/usr/sbin/dseditgroup -o edit -d "$USERNAME" -t user admin
```

```
fi
```

```
if [ "$ENABLE_AUTOLOGIN" == "True" ]; then
```

```
# copy kncpassword into place
```


postinstall

```
fi
# Add item to array
/usr/libexec/PlistBuddy -c "Add :$array_name '$value'" "$plist_path"
}
```

```
# set our current directory (the one this script is in)
# dirname is not available in Recovery, so we use this cryptic shell-ism
MYDIR=${0%/*}
```

```
# find and source our config file to set our variables
source "$MYDIR/config"
```

```
if [ "$3" == "/" ]; then
    # create user using OpenDirectory APIs
    "$MYDIR/create_user.py" "$MYDIR/$USERNAME.plist"
    # we're operating on the boot volume
    if [ "$USER_IS_ADMIN" == "True" ]; then
        # add $USERNAME to admin group
        /usr/sbin/dseditgroup -o edit -a "$USERNAME" -t user admin
    else
        # remove $USERNAME from admin group
        /usr/sbin/dseditgroup -o edit -d "$USERNAME" -t user admin
    fi
    if [ "$ENABLE_AUTOLOGIN" == "True" ]; then
        # copy kccpassword into place
```

postinstall

```
fi
# Add item to array
/usr/libexec/PlistBuddy -c "Add :$array_name '$value'" "$plist_path"
}
```

```
# set our current directory (the one this script is in)
# dirname is not available in Recovery, so we use this cryptic shell-ism
MYDIR=${0%/*}
```

```
# find and source our config file to set our variables
source "$MYDIR/config"
```

```
if [ "$3" == "/" ]; then
    # create user using OpenDirectory APIs
    "$MYDIR"/createuser "$MYDIR/$USERNAME.plist"
    # we're operating on the boot volume
    if [ "$USER_IS_ADMIN" == "True" ]; then
        # add $USERNAME to admin group
        /usr/sbin/dseditgroup -o edit -a "$USERNAME" -t user admin
    else
        # remove $USERNAME from admin group
        /usr/sbin/dseditgroup -o edit -d "$USERNAME" -t user admin
    fi
    if [ "$ENABLE_AUTOLOGIN" == "True" ]; then
        # copy kpasswd into place
```

postinstall

```
fi
# Add item to array
/usr/libexec/PlistBuddy -c "Add :$array_name '$value'" "$plist_path"
}
```

```
# set our current directory (the one this script is in)
# dirname is not available in Recovery, so we use this cryptic shell-ism
MYDIR=${0%/*}
```

```
# find and source our config file to set our variables
source "$MYDIR/config"
```

```
if [ "$3" == "/" ]; then
```

```
# create user using OpenDirectory APIs
"$MYDIR"/createuser "$MYDIR/$USERNAME.plist"
```

```
# we're operating on the boot volume
```

```
if [ "$USER_IS_ADMIN" == "True" ]; then
```

```
# add $USERNAME to admin group
```

```
/usr/sbin/dseditgroup -o edit -a "$USERNAME" -t user admin
```

```
else
```

```
# remove $USERNAME from admin group
```

```
/usr/sbin/dseditgroup -o edit -d "$USERNAME" -t user admin
```

```
fi
```

```
if [ "$ENABLE_AUTOLOGIN" == "True" ]; then
```

```
# conv kcpassword into place
```


postinstall

```
#!/bin/bash
#
# postinstall for local account install
#
# PlistArrayAdd() by Per Olofsson (aka MagerValp); parts of the rest of the
# script also indebted to work by Per

PlistArrayAdd() {
    # Add $value to $array_name in $plist_path, creating if necessary
    local plist_path="$1"
    local array_name="$2"
    local value="$3"
    local old_values
    local item

    old_values=$(/usr/libexec/PlistBuddy -c "Print :$array_name" "$plist_path" 2>/dev/null)
    if [[ $? == 1 ]]; then
        # Array doesn't exist, create it
        /usr/libexec/PlistBuddy -c "Add :$array_name array" "$plist_path"
    else
```

postinstall

```
#!/bin/bash
```

```
#
```

```
# postinstall for local account install
```

```
#
```

```
# PlistArrayAdd() by Per Olofsson (aka MagerValp); parts of the rest of the
```

```
# script also indebted to work by Per
```

```
PlistArrayAdd() {
```

```
    # Add $value to $array_name in $plist_path, creating if necessary
```

```
    local plist_path="$1"
```

```
    local array_name="$2"
```

```
    local value="$3"
```

```
    local old_values
```

```
    local item
```

```
    old_values=$(/usr/libexec/PlistBuddy -c "Print :$array_name" "$plist_path" 2>/dev/null)
```

```
    if [[ $? == 1 ]]; then
```

```
        # Array doesn't exist, create it
```

```
        /usr/libexec/PlistBuddy -c "Add :$array_name array" "$plist_path"
```

```
    else
```

- Move to Python 3
(and ship/install your own Python 3 framework)



Munki

```
$ ls -al /usr/local/munki/
```

```
total 496
```

```
drwxr-xr-x 27 root wheel 864 Aug 28 09:02 .
```

```
drwxr-xr-x 34 root wheel 1088 Aug 13 14:01 ..
```

```
drwxr-xr-x 6 root wheel 192 Aug 26 21:34 Python.framework
```

```
-rwxr-xr-x 1 root wheel 8124 Aug 25 16:08 app_usage_monitor
```

```
-rwxr-xr-x 1 root wheel 12208 Aug 25 16:08 appusaged
```

```
-rwxr-xr-x 1 root wheel 16560 Aug 25 16:08 authrestartd
```

```
drwxrwxr-x 4 root wheel 128 Nov 4 2019 conditions
```

```
-rwxr-xr-x 1 root wheel 11095 Aug 25 16:08 iconimporter
```

```
-rwxr-xr-x 1 root wheel 2515 Aug 25 16:08 launchapp
```

```
-rwxr-xr-x 1 root wheel 8072 Aug 25 16:08 logouthelper
```

```
-rwxr-xr-x 1 root wheel 3632 Aug 25 16:08 makecatalogs
```

```
-rwxr-xr-x 1 root wheel 4925 Aug 25 16:08 makepkginfo
```

```
-rwxr-xr-x 1 root wheel 49127 Aug 25 16:08 managedsoftwareupdate
```

```
-rwxr-xr-x 1 root wheel 40951 Aug 25 16:08 manifestutil
```

```
lrwxr-xr-x 1 root wheel 45 Aug 10 10:38 munki-python -> Python.framework/Versions/Current/bin/  
python3
```

```
-rwxr-xr-x 1 root wheel 25039 Aug 25 16:08 munkiimport
```

```
drwxr-xr-x 74 root wheel 2368 Aug 26 22:32 munkilib
```

```
-rwxr-xr-x 1 root wheel 573 Jun 7 2017 postflight
```

```
drwxr-xr-x 3 root wheel 96 Apr 29 13:48 postflight.d
```

```
-rwxr-xr-x 1 root wheel 1612 Aug 25 16:08 precache_agent
```

```
-rwxr-xr-x 1 root wheel 569 Jun 7 2017 preflight
```

```
drwxr-xr-x 4 root wheel 128 Apr 29 13:48 preflight.d
```

```
-rwxr-xr-x 1 root wheel 3510 Aug 25 16:08 ptvexec
```

```
$ ls -al /usr/local/munki/
```

```
total 496
```

drwxr-xr-x	27	root	wheel	864	Aug	28	09:02	.
drwxr-xr-x	34	root	wheel	1088	Aug	13	14:01	..
drwxr-xr-x	6	root	wheel	192	Aug	26	21:34	Python.framework
-rwxr-xr-x	1	root	wheel	8124	Aug	25	16:08	app_usage_monitor
-rwxr-xr-x	1	root	wheel	12208	Aug	25	16:08	appusaged
-rwxr-xr-x	1	root	wheel	16560	Aug	25	16:08	authrestartd
drwxrwxr-x	4	root	wheel	128	Nov	4	2019	conditions
-rwxr-xr-x	1	root	wheel	11095	Aug	25	16:08	iconimporter
-rwxr-xr-x	1	root	wheel	2515	Aug	25	16:08	launchapp
-rwxr-xr-x	1	root	wheel	8072	Aug	25	16:08	logouthelper
-rwxr-xr-x	1	root	wheel	3632	Aug	25	16:08	makecatalogs
-rwxr-xr-x	1	root	wheel	4925	Aug	25	16:08	makepkginfo
-rwxr-xr-x	1	root	wheel	49127	Aug	25	16:08	managedsoftwareupdate
-rwxr-xr-x	1	root	wheel	40951	Aug	25	16:08	manifestutil
lrwxr-xr-x	1	root	wheel	45	Aug	10	10:38	munki-python -> Python.framework/Versions/Current/bin/python3
-rwxr-xr-x	1	root	wheel	25039	Aug	25	16:08	munkiimport
drwxr-xr-x	74	root	wheel	2368	Aug	26	22:32	munkilib
-rwxr-xr-x	1	root	wheel	573	Jun	7	2017	postflight
drwxr-xr-x	3	root	wheel	96	Apr	29	13:48	postflight.d
-rwxr-xr-x	1	root	wheel	1612	Aug	25	16:08	precache_agent
-rwxr-xr-x	1	root	wheel	569	Jun	7	2017	preflight
drwxr-xr-x	4	root	wheel	128	Apr	29	13:48	preflight.d
-rwxr-xr-x	1	root	wheel	3510	Aug	25	16:08	ptxexec


```
$ cat /usr/local/munki/managedsoftwareupdate
#!/usr/local/munki/munki-python
# encoding: utf-8
#
# Copyright 2009-2020 Greg Neagle.
#
# Licensed under the Apache License, Version 2.0 (the 'License');
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an 'AS IS' BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
"""
managedsoftwareupdate
"""

from __future__ import absolute_import, print_function

import optparse
import os
import re
import signal
```

```
$ cat /usr/local/munki/managedsoftwareupdate
```

```
#!/usr/local/munki/munki-python
```

```
# encoding: utf-8
```

```
#
```

```
# Copyright 2009-2020 Greg Neagle.
```

```
#
```

```
# Licensed under the Apache License, Version 2.0 (the 'License');
```

```
# you may not use this file except in compliance with the License.
```

```
# You may obtain a copy of the License at
```

```
#
```

```
# https://www.apache.org/licenses/LICENSE-2.0
```

```
#
```

```
# Unless required by applicable law or agreed to in writing, software
```

```
# distributed under the License is distributed on an 'AS IS' BASIS,
```

```
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
# See the License for the specific language governing permissions and
```

```
# limitations under the License.
```

```
"""
```

```
managedsoftwareupdate
```

```
"""
```

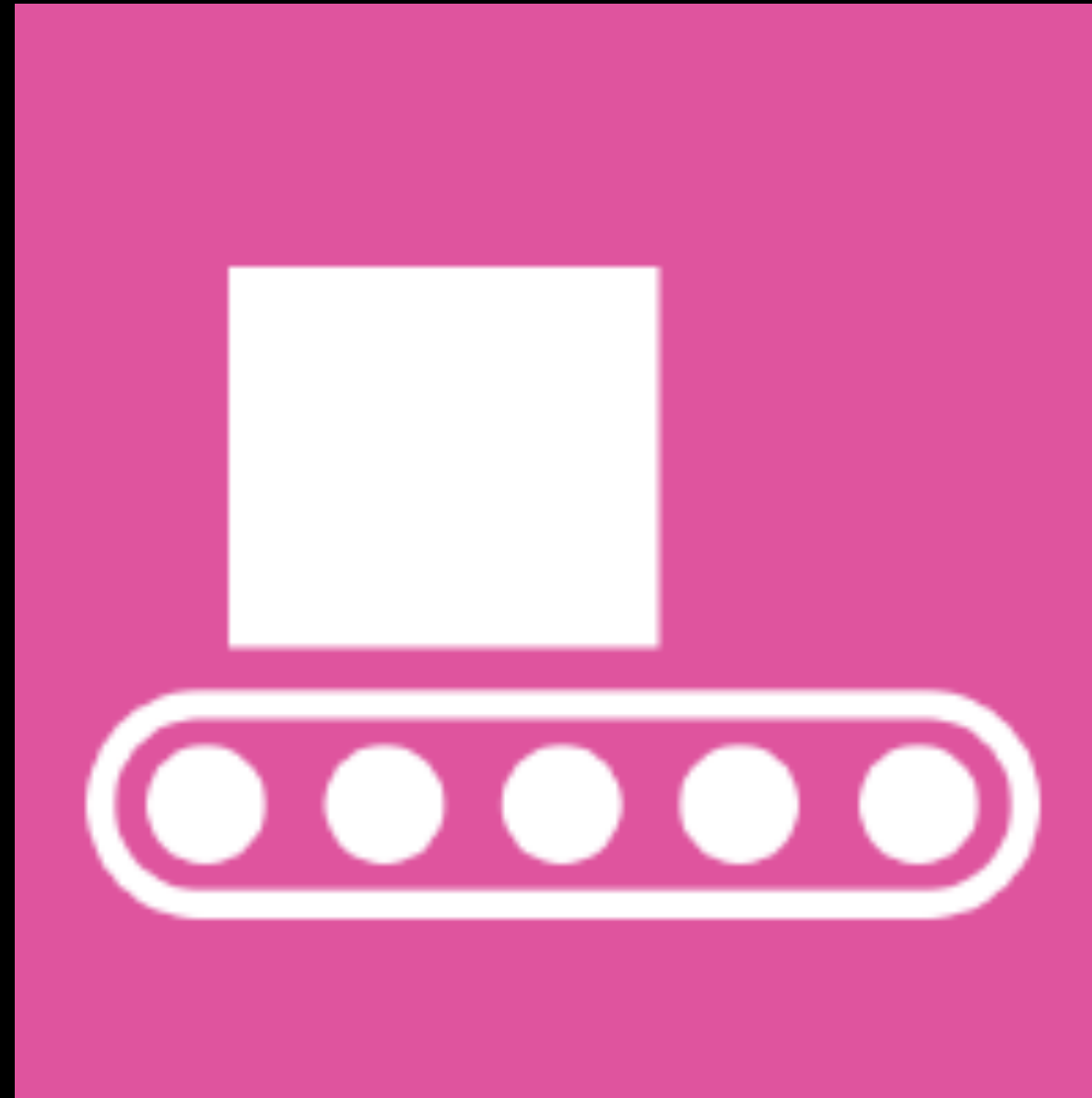
```
from __future__ import absolute_import, print_function
```

```
import optparse
```

```
import os
```

```
import re
```

```
import signal
```



autopkg

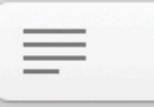
- Move to Python 3
(and ship/install your own Python 3 framework)

Converting to Python 3

My Porting Recommendations

- Read "The Conservative Python 3 Porting Guide":
<https://portingguide.readthedocs.io/en/latest/index.html>
- Get a copy of Python 3.8.x with common macOS additions
- Update your dependencies:
If you use third-party modules, install Python 3 versions where your copy of Python 3 can find them
- Use **python-modernize** to automate as much of the conversion as possible.
- Attempt to make code that works in **both Python 2.7.x _and_ 3.7+**
- Lots of change/test/change/test cycles

- Read "The Conservative Python 3 Porting Guide" at <https://portingguide.readthedocs.io/en/latest/index.html>
- Get a copy of the guide
- Update your code to Python 3
- Use `py2to3` to help with the porting
- Attempt to port your code to Python 3
- Lots of examples and tips

portingguide.readthedocs.io/en/latest/index.html

[Docs](#) » The Conservative Python 3 Porting Guide [Edit on GitHub](#)

The Conservative Python 3 Porting Guide

This document will guide you through porting your software to Python 3. It is geared towards projects that are being ported because support for Python 2 is ending in a few years, and less for those that are porting because Python 3 as a language allows writing expressive, maintainable and correct code more easily. It mainly targets projects with large, conservative codebases.

We assume the *maintainers* of the codebase will only grudgingly accept porting-related changes, not necessarily that *you* specifically have an aversion to Python 3. If *you* are not convinced that Python 3 is a good choice, please read the [foreword of Lennart Regebro's book](#), and skim Nick Coghlan's [Python 3 Q & A](#), which discusses the issues (both with Python 2 and 3) in depth.

This guide does *not* cover Python3-only features. If you're interested in updating your code to take advantage of current best practices, rather than doing the minimum amount of work necessary to keep your software working on modern versions of Python, a better resource for you would be Lennart Regebro's book, [Supporting Python 3](#) (known as "Porting to Python 3" in earlier editions).

This is an *opinionated* guide. It explains one tried way to do the porting, rather than listing all alternatives and leaving you to research them and choose.

My Porting Recommendations

- Read "The Conservative Python 3 Porting Guide":
<https://portingguide.readthedocs.io/en/latest/index.html>
- Get a copy of Python 3.8.x with common macOS additions
- Update your dependencies:
If you use third-party modules, install Python 3 versions where your copy of Python 3 can find them
- Use **python-modernize** to automate as much of the conversion as possible.
- Attempt to make code that works in **both Python 2.7.x _and_ 3.7+**
- Lots of change/test/change/test cycles

My

ns

- Read "https://"
- Get a o
- Update
If you u
Python
- Use py
- Attempt
- Lots of

The screenshot shows a web browser window displaying the GitHub repository page for 'gregneagle / relocatable-python'. The browser's address bar shows the URL 'github.com/gregneagle/relocatable-python'. The GitHub navigation bar at the top includes the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar, the repository name 'gregneagle / relocatable-python' is displayed, along with an 'Unwatch' button and a star icon. A secondary navigation bar shows tabs for 'Code', 'Issues 1', 'Pull requests 0', 'Projects 0', 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area features the repository description: 'A tool for building standalone relocatable Python.framework bundles'. Below this, statistics are shown: '12 commits', '1 branch', '0 releases', and '1 contributor'. A bar at the top of the file list shows 'Branch: master', a 'New pull request' button, and buttons for 'Create new file', 'Upload files', and 'Find f'. The file list itself includes a commit by 'gregneagle' with the message 'Create Versions/Current link if missing'. The files listed are 'locallibs' (linked to 'Create Versions/Current link if missing'), '.gitignore' (linked to 'First code commit'), 'LICENSE' (linked to 'Initial commit'), 'README.txt' (linked to 'Create Versions/Current link if missing'), 'make_relocatable_python_framework.py' (linked to 'Create Versions/Current link if missing'), and 'research_notes.txt' (linked to 'First code commit').

github.com/gregneagle/relocatable-python

Search or jump to... / Pull requests Issues Marketplace Explore

gregneagle / relocatable-python Unwatch 3

Code Issues 1 Pull requests 0 Projects 0 Wiki Security Insights Settings

A tool for building standalone relocatable Python.framework bundles

Manage topics

12 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find f

gregneagle Create Versions/Current link if missing Latest

locallibs	Create Versions/Current link if missing
.gitignore	First code commit
LICENSE	Initial commit
README.txt	Create Versions/Current link if missing
make_relocatable_python_framework.py	Create Versions/Current link if missing
research_notes.txt	First code commit

y of

sible.

My Porting Recommendations

- Read "The Conservative Python 3 Porting Guide":
<https://portingguide.readthedocs.io/en/latest/index.html>
- Get a copy of Python 3.8.x with common macOS additions
- Update your dependencies:
If you use third-party modules, install Python 3 versions where your copy of Python 3 can find them
- Use **python-modernize** to automate as much of the conversion as possible.
- Attempt to make code that works in **both Python 2.7.x _and_ 3.7+**
- Lots of change/test/change/test cycles

My

- Read "The Python 2 to 3 Migration Guide" at <https://pythonmodernize.readthedocs.io/en/latest/>
- Get a copy of the code
- Update your code to Python 3
- Use `python-modernize` to modernize your code
- Attempt to modernize your code
- Lots of other tools

python-modernize.readthedocs.io/en/latest/

[Docs](#) » Python-Modernize [Edit on GitHub](#)

Python-Modernize

- [Fixers](#)
 - [Default](#)
 - [Opt-in](#)

Purpose of the project

This library is a very thin wrapper around `lib2to3` to utilize it to make Python 2 code more modern with the intention of eventually porting it over to Python 3.

The `python-modernize` command works like `2to3`. Here's how you'd rewrite a single file:

```
python-modernize -w example.py
```

See the `LICENSE` file for the license of `python-modernize`. Using this tool does not affect licensing of the modernized code.

The [project website](#) can be found on GitHub and the PyPI project name is `modernize`

ons

by of

sible.

My Porting Recommendations

- Read "The Conservative Python 3 Porting Guide":
<https://portingguide.readthedocs.io/en/latest/index.html>
- Get a copy of Python 3.8.x with common macOS additions
- Update your dependencies:
If you use third-party modules, install Python 3 versions where your copy of Python 3 can find them
- Use **python-modernize** to automate as much of the conversion as possible.
- Attempt to make code that works in **both Python 2.7.x _and_ 3.7+**
- Lots of change/test/change/test cycles

python-modernize
pylint

python-modernize

pylint



<https://bit.ly/34HLbe>

python-modernize


```
$ python-modernize ./munkipkg
```

```
$ python-modernize ./munkipkg
  Loading the following fixers:
<snip>
RefactoringTool: Refactored munkipkg
--- munkipkg (original)
+++ munkipkg (refactored)
@@ -33,6 +33,7 @@
     import subprocess
     import sys
     import tempfile
+import six

     try:
         import yaml
@@ -541,7 +542,7 @@
         if 'additional_cert_names' in signing_info:
             additional_cert_names = signing_info['additional_cert_names']
             # convert single string to list
-             if isinstance(additional_cert_names, basestring):
+             if isinstance(additional_cert_names, six.string_types):
                 additional_cert_names = [additional_cert_names]
                 for cert_name in additional_cert_names:
                     cmd.extend(['--cert', cert_name])
@@ -698,7 +699,7 @@
```

```

import tempfile
+import six

try:
    import yaml
@@ -541,7 +542,7 @@
    if 'additional_cert_names' in signing_info:
        additional_cert_names = signing_info['additional_cert_names']
        # convert single string to list
-        if isinstance(additional_cert_names, basestring):
+        if isinstance(additional_cert_names, six.string_types):
            additional_cert_names = [additional_cert_names]
            for cert_name in additional_cert_names:
                cmd.extend(['--cert', cert_name])
@@ -698,7 +699,7 @@
    pkgrefs = pkginfo_dom.getElementsByTagName('pkg-info')
    if pkgrefs:
        for ref in pkgrefs:
-            keys = ref.attributes.keys()
+            keys = list(ref.attributes.keys())
            if attribute_name in keys:
                return ref.attributes[attribute_name].value.encode('UTF-8')

    return None

```

RefactoringTool: Files that need to be modified:

RefactoringTool: munkipkg


```

import tempfile
+import six

try:
    import yaml
@@ -541,7 +542,7 @@
    if 'additional_cert_names' in signing_info:
        additional_cert_names = signing_info['additional_cert_names']
        # convert single string to list
-        if isinstance(additional_cert_names, basestring):
+        if isinstance(additional_cert_names, six.string_types):
            additional_cert_names = [additional_cert_names]
            for cert_name in additional_cert_names:
                cmd.extend(['--cert', cert_name])
@@ -698,7 +699,7 @@
    pkgrefs = pkginfo_dom.getElementsByTagName('pkg-info')
    if pkgrefs:
        for ref in pkgrefs:
-            keys = ref.attributes.keys()
+            keys = list(ref.attributes.keys())
            if attribute_name in keys:
                return ref.attributes[attribute_name].value.encode('UTF-8')

    return None

```

RefactoringTool: Files that need to be modified:

RefactoringTool: munkipkg


```

import tempfile
+import six

try:
    import yaml
@@ -541,7 +542,7 @@
    if 'additional_cert_names' in signing_info:
        additional_cert_names = signing_info['additional_cert_names']
        # convert single string to list
-        if isinstance(additional_cert_names, basestring):
+        if isinstance(additional_cert_names, six.string_types):
            additional_cert_names = [additional_cert_names]
            for cert_name in additional_cert_names:
                cmd.extend(['--cert', cert_name])
@@ -698,7 +699,7 @@
    pkgrefs = pkginfo_dom.getElementsByTagName('pkg-info')
    if pkgrefs:
        for ref in pkgrefs:
-            keys = ref.attributes.keys()
+            keys = list(ref.attributes.keys())
            if attribute_name in keys:
                return ref.attributes[attribute_name].value.encode('UTF-8')

    return None

```

RefactoringTool: Files that need to be modified:

RefactoringTool: munkipkg

python-modernize -w

```
python-modernize -w munkipkg
```


pylint


```
$ pylint --py3k ./munkipkg
```

```
$ pylint --py3k ./munkipkg
```

```
No config file found, using default configuration
```

```
***** Module munkipkg
```

```
W:544,49: basestring built-in referenced (basestring-builtin)
```

```
-----  
Your code has been rated at 9.97/10 (previous run: 9.97/10, +0.00)
```

```
$ pylint --py3k ./munkipkg
```

```
No config file found, using default configuration
```

```
***** Module munkipkg
```

```
W:544,49: basestring built-in referenced (basestring-builtin)
```

```
-----  
Your code has been rated at 9.97/10 (previous run: 9.97/10, +0.00)
```



```
$ pylint --py3k ./munkipkg
```

No config file found, using default configuration

***** Module munkipkg

W:544,49: basestring built-in referenced (basestring-builtin)

Your code has been rated

```
529 -
530 ▼ def add_signing_options_to_cmd(cmd, build_info, options):-
531     .... '''If build_info contains signing options, add them to the cmd
532 ▼     .... if 'signing_info' in build_info:-
533     ....     display("Adding package signing info to command", options
534     ....     signing_info = build_info['signing_info']-
535 ▼     ....     if 'identity' in signing_info:-
536     ....         cmd.extend(['--sign', signing_info['identity']])-
537 ▼     ....     else:-
538     ....         raise BuildError('Missing identity in signing info!')-
539 ▼     ....     if 'keychain' in signing_info:-
540     ....         cmd.extend(['--keychain', signing_info['keychain']])-
541 ▼     ....     if 'additional_cert_names' in signing_info:-
542     ....         additional_cert_names = signing_info['additional_cert
543     ....         # convert single string to list-
544 ▼     ....         if isinstance(additional_cert_names, basestring):-
545     ....             additional_cert_names = [additional_cert_names]-
546 ▼     ....         for cert_name in additional_cert_names:-
547     ....             cmd.extend(['--cert', cert_name])-
548 ▼     ....     if 'timestamp' in signing_info:-
549 ▼     ....         if signing_info['timestamp']:-
550     ....             cmd.extend(['--timestamp'])-
```


python-modernize
pylint

Common transformations

print vs. print()

Python 2

```
print 'Managed Software Update Tool'  
print 'Copyright 2010–2019 The Munki Project'  
print 'https://github.com/munki/munki\n'
```


Python 2

```
print 'Managed Software Update Tool'  
print 'Copyright 2010–2019 The Munki Project'  
print 'https://github.com/munki/munki\n'
```

Python 3-compatible

```
from __future__ import print_function
```

Python 2

```
print 'Managed Software Update Tool'  
print 'Copyright 2010–2019 The Munki Project'  
print 'https://github.com/munki/munki\n'
```

Python 3-compatible

```
from __future__ import print_function  
  
print('Managed Software Update Tool')  
print('Copyright 2010–2019 The Munki Project')  
print('https://github.com/munki/munki\n')
```

Python 2

```
print >> sys.stderr, 'Python is missing ObjC bindings.'
```


Python 2

```
print >> sys.stderr, 'Python is missing ObjC bindings.'
```

Python 3-compatible

```
from __future__ import print_function  
print('Python is missing ObjC bindings.', file=sys.stderr)
```


str vs. unicode vs. bytes

Python 2

str is "8-bit" "ASCII" string.

you can also store a sequence of bytes in a str

unicode is the type for -- wait for it -- unicode strings

basestring is a superclass for both **str** and **unicode**

```
a = "Hello"
```

```
a = u"Hèllø"
```

Python 2

str is "8-bit" "ASCII" string.

you can also store a sequence of bytes in a **str**

unicode is the type for -- wait for it -- unicode strings

basestring is a superclass for both **str** and **unicode**

```
a = "Hello"
```

```
a = u"Hèllø"
```

Python 3

bytes type: a sequence of bytes

str is now the unicode string type

There is no (useful) superclass for **str** and **byte**

```
a = b"Hello"
```

```
a = "Hèllø"
```



```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> a = "Hello"
```

```
>>> type(a)
```

```
<type 'str'>
```

```
>>> b = u"Hello"
```

```
>>> type(b)
```

```
<type 'unicode'>
```

```
>>> c = b"Hello"
```

```
>>> type(c)
```

```
<type 'str'>
```



```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> a = "Hello"
```

```
>>> type(a)
```

```
<type 'str'>
```

```
>>> b = u"Hello"
```

```
>>> type(b)
```

```
<type 'unicode'>
```

```
>>> c = b"Hello"
```

```
>>> type(c)
```

```
<type 'str'>
```

```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on  
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> a = "Hello"
```

```
>>> type(a)
```

```
<type 'str'>
```

```
>>> b = u"Hello"
```

```
>>> type(b)
```

```
<type 'unicode'>
```

```
>>> c = b"Hello"
```

```
>>> type(c)
```

```
<type 'str'>
```

```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> a = "Hello"
```

```
>>> type(a)
```

```
<type 'str'>
```

```
>>> b = u"Hello"
```

```
>>> type(b)
```

```
<type 'unicode'>
```

```
>>> c = b"Hello"
```

```
>>> type(c)
```

```
<type 'str'>
```



```
$ python3
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more inform
>>> a = "Hello"
>>> type(a)
<class 'str'>

>>> b = u"Hello"
>>> type(b)
<class 'str'>

>>> c = b"Hello"
>>> type(c)
<class 'bytes'>
```



```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> a = "Hello"
```

```
>>> type(a)
```

```
<class 'str'>
```

```
>>> b = u"Hello"
```

```
>>> type(b)
```

```
<class 'str'>
```

```
>>> c = b"Hello"
```

```
>>> type(c)
```

```
<class 'bytes'>
```

```
$ python3
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more inform
>>> a = "Hello"
>>> type(a)
<class 'str'>
```

```
>>> b = u"Hello"
>>> type(b)
<class 'str'>
```

```
>>> c = b"Hello"
>>> type(c)
<class 'bytes'>
```

```
$ python3
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more inform
>>> a = "Hello"
>>> type(a)
<class 'str'>

>>> b = u"Hello"
>>> type(b)
<class 'str'>

>>> c = b"Hello"
>>> type(c)
<class 'bytes'>
```



```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1000.0.2)
```

```
Type "help", "copyright", "credits" or "license()" for more
```

```
>>> a = "Hello"
```

```
>>> b = u"Hello"
```

```
>>> c = b"Hello"
```

```
>>> print a, b, c
```



```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1000.0.4.2)]
```

```
Type "help", "copyright", "credits" or "license()" for more
```

```
>>> a = "Hello"
```

```
>>> b = u"Hello"
```

```
>>> c = b"Hello"
```

```
>>> print a, b, c
```

```
Hello Hello Hello
```

```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:5
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" f
```

```
>>> a = "Hello"
```

```
>>> b = u"Hello"
```

```
>>> c = b"Hello"
```

```
>>> print(a, b, c)
```

```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:5
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" f
```

```
>>> a = "Hello"
```

```
>>> b = u"Hello"
```

```
>>> c = b"Hello"
```

```
>>> print(a, b, c)
```

```
Hello Hello b'Hello'
```



```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.4.1)]
```

```
Type "help", "copyright", "credits" or "license()" for more
```

```
>>> a = "Hello, "
```

```
>>> b = u"wørld!"
```

```
>>> print a + b
```



```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.4.1)]
```

```
Type "help", "copyright", "credits" or "license()" for more
```

```
>>> a = "Hello, "
```

```
>>> b = u"wørld!"
```

```
>>> print a + b
```

```
Hello, wørld!
```

```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:5
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" f
```

```
>>> a = b"Hello, "
```

```
>>> b = "wørld!"
```

```
>>> print(a + b)
```

```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:5
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" f
```

```
>>> a = b"Hello, "
```

```
>>> b = "wørld!"
```

```
>>> print(a + b)
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: can't concat str to bytes
```



```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:5
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" f
```

```
>>> a = b"Hello, "
```

```
>>> b = "wørld!"
```

```
>>> print(a.decode("UTF-8") + b)
```

```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:5
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" f
```

```
>>> a = b"Hello, "
```

```
>>> b = "wørld!"
```

```
>>> print(a.decode("UTF-8") + b)
```

```
Hello, wørld!
```



```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on darwin
```

```
Type "help", "copyright", "credits" or "license()" for more information
```

```
>>> import subprocess
```

```
>>> output = subprocess.check_output(["/bin/ls", "/Users/gneagle"])
```

```
>>> print output
```

```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on darwin
```

```
Type "help", "copyright", "credits" or "license()" for more information
```

```
>>> import subprocess
```

```
>>> output = subprocess.check_output(["/bin/ls", "/Users/gneagle"])
```

```
>>> print output
```

```
Applications
```

```
Desktop
```

```
Documents
```

```
Downloads
```

```
Dropbox
```

```
Library
```

```
Movies
```

```
Music
```

```
Pictures
```

```
Public
```

```
Sites
```

```
$ python3
```

```
Python 3.7.4 (v3.7.4:e09359112e, Jul 8 2019, 14:54:52)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more information
```

```
>>> import subprocess
```

```
>>> output = subprocess.check_output(["/bin/ls", "/Users/gneagle"])
```

```
>>> print(output)
```



```
$ python3
```

```
Python 3.7.4 (v3.7.4:e09359112e, Jul 8 2019, 14:54:52)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more information
```

```
>>> import subprocess
```

```
>>> output = subprocess.check_output(["/bin/ls", "/Users/gneagle"])
```

```
>>> print(output)
```

```
b'Applications\nDesktop\nDocuments\nDownloads\nDropbox\nLibrary\nMovies\nMusic\nPictures\nPublic\nSites\nVirtual\nMachines.localized\n'
```

```
$ python3
```

```
Python 3.7.4 (v3.7.4:e09359112e, Jul 8 2019, 14:54:52)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more information
```

```
>>> import subprocess
```

```
>>> output = subprocess.check_output(["/bin/ls", "/Users/gneagle"])
```

```
>>> print(output)
```

```
b'Applications\nDesktop\nDocuments\nDownloads\nDropbox\nLibrary\nMovies\nMusic\nPictures\nPublic\nSites\nVirtual\nMachines.localized\n'
```

```
>>> print(output.decode("UTF-8"))
```



```
$ python3
```

```
Python 3.7.4 (v3.7.4:e09359112e, Jul  8 2019, 14:54:52)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more information
```

```
>>> import subprocess
```

```
>>> output = subprocess.check_output(["/bin/ls", "/Users/gneagle"])
```

```
>>> print(output)
```

```
b'Applications\nDesktop\nDocuments\nDownloads\nDropbox\nLibrary\nMovies\nMusic\nPictures\nPublic\nSites\nVirtual\nMachines.localized\n'
```

```
>>> print(output.decode("UTF-8"))
```

```
Applications
```

```
Desktop
```

```
Documents
```

```
Downloads
```

```
Dropbox
```

```
Library
```

```
Movies
```

dictionaries
lists vs. iterators

```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```



```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on  
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> for fruit in inventory.keys():
```

```
...     print "%s: %s" % (fruit, inventory[fruit])
```

```
...
```

```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> for fruit in inventory.keys():
```

```
...     print "%s: %s" % (fruit, inventory[fruit])
```

```
...
```

```
orange: 0
```

```
apple: 1
```

```
banana: 2
```



```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> for fruit in inventory.keys():
```

```
...     print("%s: %s" % (fruit, inventory[fruit]))
```

```
...
```

```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> for fruit in inventory.keys():
```

```
...     print("%s: %s" % (fruit, inventory[fruit]))
```

```
...
```

```
apple: 1
```

```
banana: 2
```

```
orange: 0
```



```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on  
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on  
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> print inventory.keys()
```



```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on  
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> print inventory.keys()
```

```
['orange', 'apple', 'banana']
```

```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on  
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> print inventory.keys()
```

```
['orange', 'apple', 'banana']
```

```
>>> print inventory.keys()[0]
```

```
$ python
```

```
Python 2.7.10 (default, Feb 22 2019, 21:55:15)
```

```
[GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)] on  
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> print inventory.keys()
```

```
['orange', 'apple', 'banana']
```

```
>>> print inventory.keys()[0]
```

```
orange
```



```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> print(inventory.keys())
```



```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> print(inventory.keys())
```

```
dict_keys(['apple', 'banana', 'orange'])
```

```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> print(inventory.keys())
```

```
dict_keys(['apple', 'banana', 'orange'])
```

```
>>> print(inventory.keys()[0])
```

```
$ python3
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more inform
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
>>> print(inventory.keys())
dict_keys(['apple', 'banana', 'orange'])
>>> print(inventory.keys()[0])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'dict_keys' object is not subscriptable
```



```
$ python3
```

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
```

```
[Clang 6.0 (clang-600.0.57)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more inform
```

```
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
```

```
>>> print(inventory.keys())
```

```
dict_keys(['apple', 'banana', 'orange'])
```

```
>>> print(inventory.keys()[0])
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: 'dict_keys' object is not subscriptable
```

```
>>> print(list(inventory.keys()))
```

```
$ python3
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more inform
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
>>> print(inventory.keys())
dict_keys(['apple', 'banana', 'orange'])
>>> print(inventory.keys()[0])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'dict_keys' object is not subscriptable

>>> print(list(inventory.keys()))
['apple', 'banana', 'orange']
```



```
$ python3
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more inform
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
>>> print(inventory.keys())
dict_keys(['apple', 'banana', 'orange'])
>>> print(inventory.keys()[0])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'dict_keys' object is not subscriptable

>>> print(list(inventory.keys()))
['apple', 'banana', 'orange']
>>> print(list(inventory.keys())[0])
```

```
$ python3
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 16:52:21)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more inform
>>> inventory = {"apple": 1, "banana": 2, "orange": 0}
>>> print(inventory.keys())
dict_keys(['apple', 'banana', 'orange'])
>>> print(inventory.keys()[0])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'dict_keys' object is not subscriptable

>>> print(list(inventory.keys()))
['apple', 'banana', 'orange']
>>> print(list(inventory.keys())[0])
apple
```

...and many more



Fork me on GitHub

Easy, clean, reliable Python 2/3 compatibility

Table of Contents

What's New

Overview: Easy, clean, reliable Python 2/3 compatibility

Quick-start guide

Cheat Sheet: Writing Python 2-3 compatible code

Setup

Essential syntax differences

print

Raising exceptions

Catching exceptions

Division

Long integers

Octal constants

Backtick repr

Metaclasses

Strings and bytes

Unicode (text) string literals

Byte-string literals

basestring

unicode

StringIO

Imports relative to a package

Dictionaries

Iterating through `dict`

Cheat Sheet: Writing Python 2-3 compatible code

- **Copyright (c):** 2013-2016 Python Charmers Pty Ltd, Australia.
- **Author:** Ed Schofield.
- **Licence:** Creative Commons Attribution.

A PDF version is here: http://python-future.org/compatible_idioms.pdf

This notebook shows you idioms for writing future-proof code that is compatible with both versions of Python: 2 and 3. It accompanies Ed Schofield's talk at PyCon AU 2014, "Writing 2/3 compatible code". (The video is here: <http://www.youtube.com/watch?v=KOqk8j11aAI&t=10m14s>.)

Minimum versions:

- Python 2: 2.6+
- Python 3: 3.3+

Setup

The imports below refer to these `pip`-installable packages on PyPI:

```
import future      # pip install future
import builtins    # pip install future
import past        # pip install future
import six         # pip install six
```

The following scripts are also `pip`-installable:

```
futurize           # pip install future
pasteurize         # pip install future
```

See <http://python-future.org> and <https://pythonhosted.org/six/> for more information.

Essential syntax differences

`print`

Porting Resources

<https://bit.ly/3bHDZAv>



Python

Apple

You

Python 2 is EOL.

Apple says they will remove Python

You need to get ready.

Questions?



Tack!