# Python Quick Start for Linux System Administrators

#### PYTHON QUICK START



**Dr Chris Brown** 



#### Why Python?



Easy to learn and to write

Interpreted - can be used interactively

Powerful built-in data types

- Strings, integers, floats
- Lists, tuples, dictionaries, sets

#### **Object-oriented**

- Many pre-defined classes
- Gives the language a "high level" feel



#### "Batteries Included"



#### **Huge collection of modules**

- Many are part of the python distribution
- Many more in your distro's repositories
- Huge number at Python Package Index pypi.python.org



## A Small Sample of Modules

Module	Description
math	Trig functions, sqrt, logarithms, exponentials
pickle	Serialise / de-serialize objects for persistent storage
random	Generate random numbers with various distributions
re	Regular expression pattern matching and substitution
string	Comprehensive string formatting
configparser	Configuration file parser
bz2, gzip	Read and write compressed files
tarfile	Read and write tar archives
datetime	Represent and manipulate dates and times

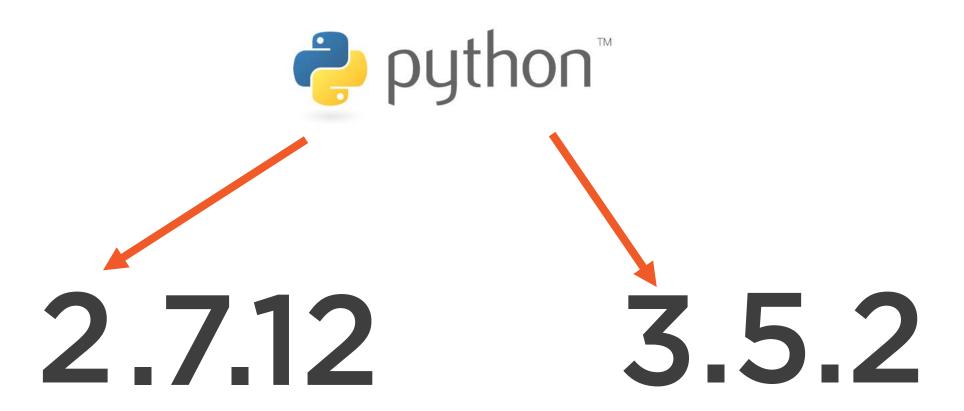


## A Small Sample of Modules

Module	Description
logging	Log message generator with various backends
optparse	Parse command-line arguments
os	Interface to operating system services
sys	Access argv, stdin, stdout, etc.
socket	Python binding for the traditional BSD socket API
http	Modules for client and server side http, and cookies
shutil	Copy / remove files and directory trees
glob	Shell-style wildcard expansion
xml	Processing of XML data

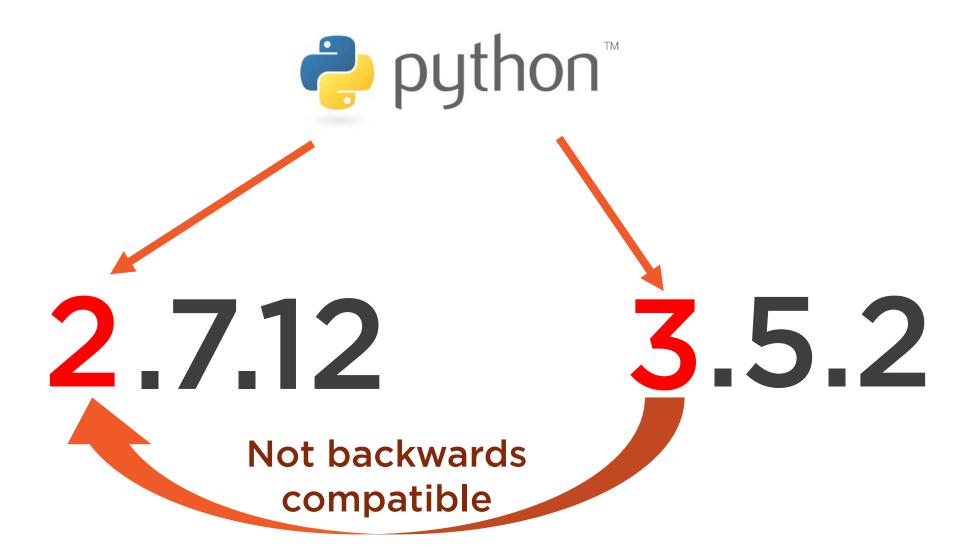


#### A Tale of Two Pythons



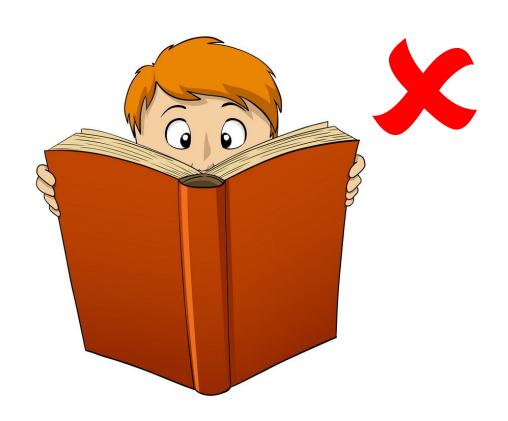


#### A Tale of Two Pythons

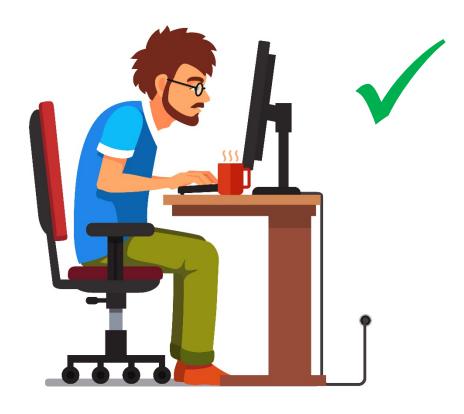




## Learning a Language



All at once



On the job



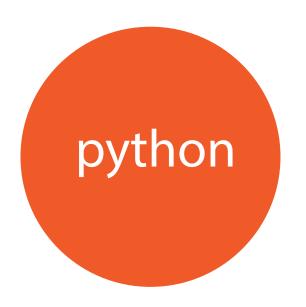
## Replacing "Little Languages"





### Replacing "Little Languages"







#### Comparing python and bash

```
#!/bin/bash
factorial()
    fac=1
    for ((n = \$1; n > 0; n--))
    do
       ((fac = fac * n))
    done
    echo $fac
fac5=$(factorial 5)
echo factorial 5 is $fac5
```

```
#!/usr/bin/python3

def factorial(n):
    fac = 1
    for x in range(1, n+1):
        fac = fac * x
    return fac

print(factorial(5))
```



#### Comparing python and bash

```
#!/bin/bash

if [ -e /etc/hosts ]
then
    echo hosts file exists
else
    echo no hosts file
fi
```

```
#!/usr/bin/python3
import os.path
if os.path.exists("/etc/hosts"):
    print("hosts file exists")
else:
    print("no hosts file")
```



#### The "Pythonic" Way

```
#!/usr/bin/python3

try:
    f = open("/etc/hosts")
      # go ahead and read the file
except FileNotFoundError:
    print("no hosts file")
```

"It is easier to ask forgiveness than permission"



#### Comparing python and bash

"Find the largest UID in the password file"

```
$ sort -n -t: -k3 /etc/passwd | tail -1 | cut -d: -f3
```

```
#!/usr/bin/python3

maxuid = 0
for line in open("/etc/passwd"):
    split = line.split(":")
    if int(split[2]) > maxuid:
        maxuid = int(split[2])

print(maxuid)
```



#### Summary



## Why python makes a good scripting language Batteries included!

- Packages to support a huge range of tasks

Replacing "Little Languages"

#### Comparing python with bash

- The use of colons and leading whitespace
- Easier to ask forgiveness than permission!



#### Coming Up ...



## Choosing your development environment

