

Introducing Python

By Bill Lubanovic

An edible recipe for Learning Python

SME: AI Nash

A Taste Of Py

- ✦ Python in the Real World
 - ✦ Used Everyday at Google, Hulu, Dropbox, Netflix and Youtube.
 - ✦ Python ranges from one-off scripts to million-line systems
- ✦ So, Why Python
 - ✦ Python is a good general-purpose language, high-level language
 - ✦ Python has a gentle learning curve when compared to other languages
- ✦ Your moment of Zen. — [import this](#)

A Taste Of Py. Pt.2

- ✦ When Not to use Python
 - ✦ Its not installed everywhere by default
 - ✦ Fast enough for most, applications but it might not be fast enough for some demanding ones. You might have an extremely demanding application and no matter what Python doesn't meet your needs
 - ✦ Currently two versions available Python 2.x vs Python 3.0

Py Ingredients



Variables, Numbers, Objects

Variables

`x = y`

Numbers

Integers
(1,2,3,4,5..)

Bases
0x10

Floats
(1.0, 2.0 etc)

Math
Functions
+, -, *, \

Objects

`str()`

`split()`

`replace()`

`join()`

`len()`

Slice
step:end:start

Py Filling

Lists, Tuples, Dictionaries, Sets

Lists

Tuples

Dictionaries

Sets

[] or list()

()

{ } or dict()

set()

append()
extend()
insert()

empty_tu
ple = ()

key:value
pair

Test for value using *in*

del
remove()
pop()

tuple vs
list
less space

keys()
values()
items()

Make bigger data structures

index()
sort()
count()

use as
keys
within

copy()

Use a set when you only
want to know if something
exists, and nothing about it

Py Crust

Code Structures

- Comment with #
- Continue with /
- Repeat with *while*
 - Skip ahead with *continue*
- Iterate with for
- Comprehensions
 - List, Dictionaries, Generators
- Functions

Py Crust pt 2

Code Structures

- Generators : Python sequence creation object
- Decorators : takes another function as input and returns another function
- Handle errors with *try and except*
- Make your own exceptions

Py Boxes

Modules, Packages, Programs

- “This textbook was organized in a hierarchy: words, sentences, paragraphs, and chapters”
- “Code has the similar bottom-up organization”:
 - data types are like words
 - statements are like sentences
 - functions are like paragraphs
 - modules are like chapters
- To allow Python applications to scale from you can organize modules into ***packages***

Oh Oh:Objects and Classes

- An **object** contains both data (variables called **attributes**) and code (functions, called **methods**)
- A **class** is like the mold that makes a box
- Define a class with **class**
 - `class Person():`

Mangle Data

- Formatting, Encoding, Decoding
 - Unicode
 - UTF-8
 - formatting with {} and ***format.***
 - *Binary Data*

Data Has to Go Somewhere

- `write()`
- `read()`, `readline()` or `readlines()`
- Close files automatically using ***with***
- Structured Text Filer
 - CSV ‘
 - XML, YAML
 - HTML
 - JSON

The Web

- Python Standard Web Libraries include:
 - Web Clients (http and urllib)
 - Web Server (Bottle, Flask , Pickle)
 - Web Services and Automation (*webbrowser*,
 - Web APIs (REST, JSON)

Systems

os = Python system functions module

Files	Directories	Processes	Time Management
open() exists() isfile() copy()	mkdir() rmdir() listdir()	subprocess	datetime isoformat() now() strptime() strftime() combine() today()
rename() link() symlink() chmod() chown()	chdir()	multiprocess	time time() ctime() struct_time localtime() gmtime() locale
abspath() realpath() remove()	glob()	terminate()	dateutil iso8601 fleming arrow

Concurrency and Networks

Concurrency

multiprocessing
threading

gevent

twisted

asyncio
redis

Message Brokers

ZeroMQ

RabbitMQ

Pub-Sub Model

Networks

TCP/IP
Sockets

DNS

gethostbyname()
getaddrinfo()

Scapy

Be a Pythonista

Find	Install	IDEs	Docs	Test	Optimiz	VCS
Python Standard Library	pip	IDLE	pylint	unittest doctest	time	Git
PyPI	apt- get	PyChar m	pyflake s	nose	algos	Mecurial
	dpkg	IPython	pep8	jenkins	Cython NumPy	
	zypp r			pdb	PyPy	

.close() .stop() Q&A

- ✦ References included within Introducing Python
 - ✦ Clone this book: www.github.com/madscheme/introducing-python
 - ✦ PyConn: www.pycon.org
 - ✦ Workshops: www.python.org/community/workshops