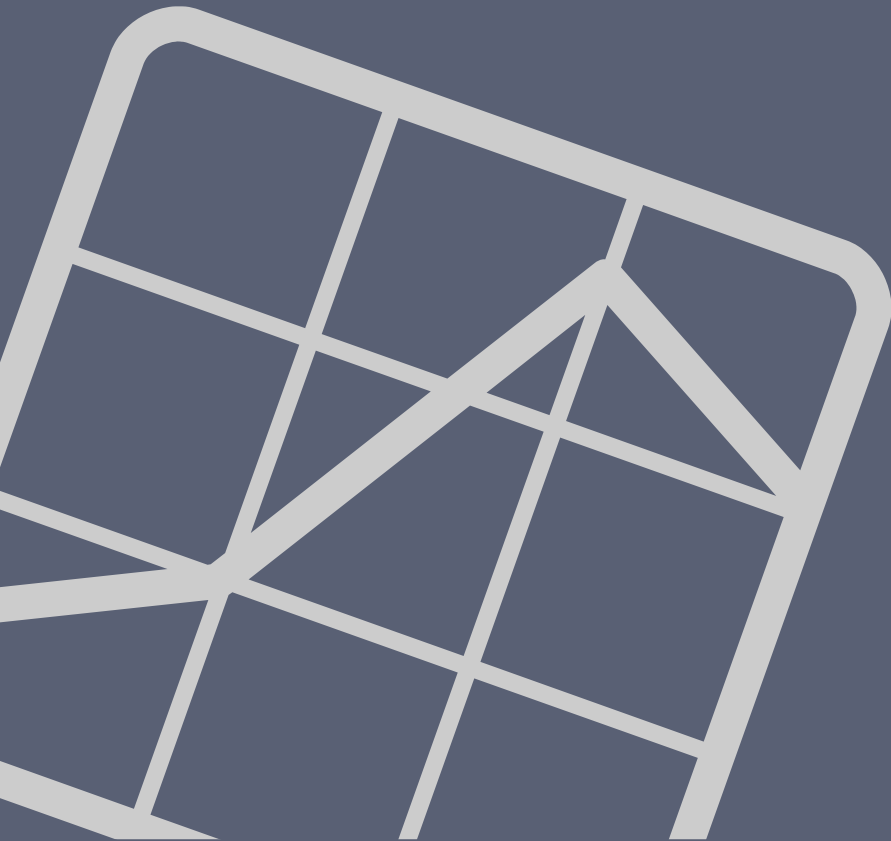


Madagascar IEES Training

# Introduction to Python

Jonathan Berry



# Overview

**Talk:** Introduction to python

**Exercises:** Python tutorial

**Exercises:** Advanced python problems

**Talk:** No more PhP - Web programming in Python

**Exercises:** Python websites

**Talk:** Javascript in Meerkat Frontend

**Exercises:** Editing Meerkat Frontend



# Where?

Meerkat Abacus

Meerkat Frontend

Meerkat API

Meerkat Hermes

Meerkat Nest

Meerkat Auth



# Why?

- Cross platform
- 4<sup>th</sup> most popular code language (After C, Java, C++)
- Very few lines do an awful lot!
- You can do almost anything you want in Python
- Stats and science speciality



# What?

A scripting language

**An interpreted language**

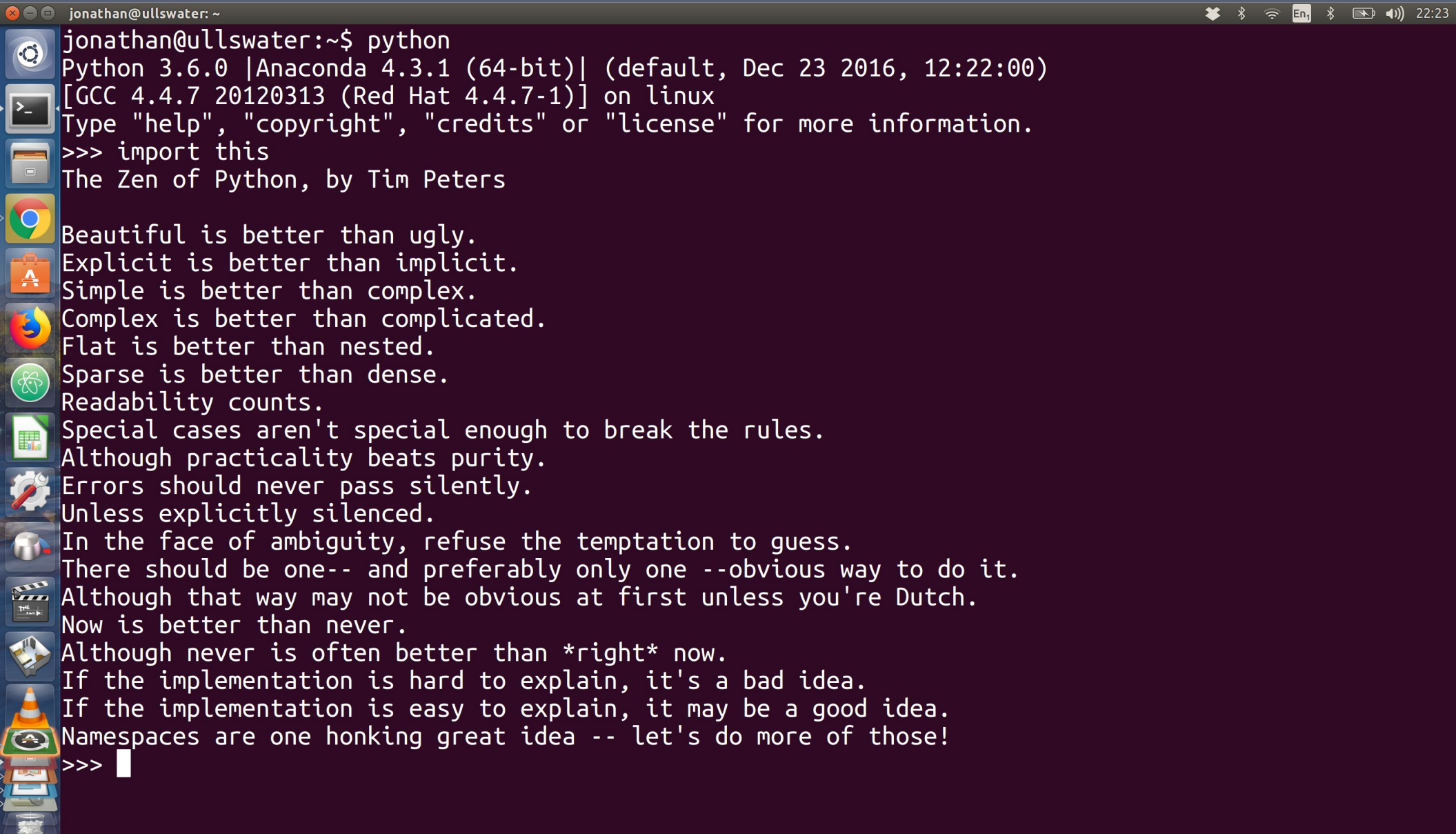
**An object orientated language**

Dynamically typed

**Functional programming support**



# How?



A terminal window titled 'jonathan@ullswater: ~' with a dark background and light text. The window shows the execution of the 'python' command, which displays the Python version (3.6.0), the environment (Anaconda 4.3.1), the compiler (GCC 4.4.7), and the OS (Linux). It then prompts the user to type 'help', 'copyright', 'credits', or 'license'. The user enters 'import this', which displays the 'Zen of Python' by Tim Peters. The Zen of Python is a list of 20 guidelines for writing Python code, such as 'Beautiful is better than ugly' and 'Explicit is better than implicit'. The terminal window has a sidebar on the left with various application icons and a top bar with system status icons and the time '22:23'.

```
jonathan@ullswater:~$ python
Python 3.6.0 |Anaconda 4.3.1 (64-bit)| (default, Dec 23 2016, 12:22:00)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>> 
```



# How?

Key features:

- Uses white-space and indentation!
- No curly braces, no semi colons.
- Can use Python console
- Can run Python files
- Statements are very simple!

```
for x in array:  
    print(x)
```



Guido van Rossum

Exercise

# A simple French Python Tutorial

[python.50webs.com](https://python.50webs.com)

