

# Python, Qt4, QGIS & InaSAFE



By Tim Sutton

Clone this talk!

`git clone git@github.com:timlinux/python-qt4-qgis-inasafe-talk.git`

# Talk outline

4 topics

- \* PyQt4
- \* QGIS
- \* InaSAFE
- \* Lessons we have learned / Python essentials

# Part 1

# PyQt4

# What is Qt?

- \* Set of C++ libraries (with Python bindings)
- \* Gui, File I/O, Networking, Web, Xml, etc.
- \* Cross-platform
- \* Android, Linux, Windows, OSX, BBerry
- \* Basis for KDE, BB 10, QGIS, Google Earth...
- \* FOSS at <http://qt.digia.com/>

# PyQt4 HelloWorld

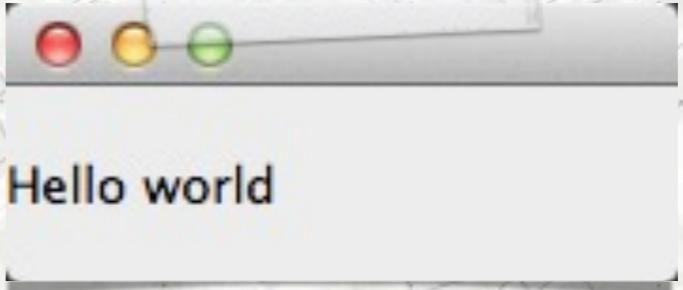
```
# coding=utf-8
"""
Example Qt4 Application
"""

__author__ = 'tim@linfiniti.com'
__revision__ = '$Format:%H$'
__date__ = '27/05/2013'
__copyright__ = 'Copyright 2012, Tim Sutton'

import sys

from PyQt4 import Qt, QtGui

if __name__ == '__main__':
    app = Qt.QApplication(sys.argv)
    label = QtGui.QLabel('Hello world')
    label.show()
    sys.exit(app.exec_())
```



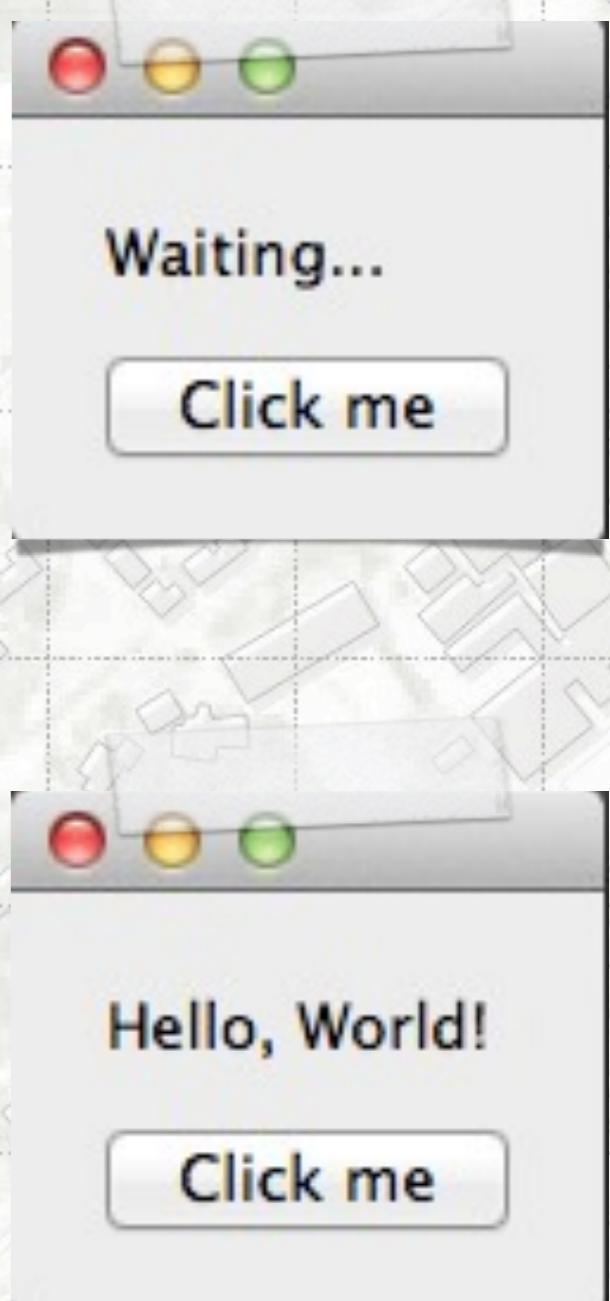
# PyQt4 Application

```
#!/usr/bin/env python
# coding=utf-8
"""Example Qt4 Application"""
import sys
from PyQt4 import Qt, QtGui

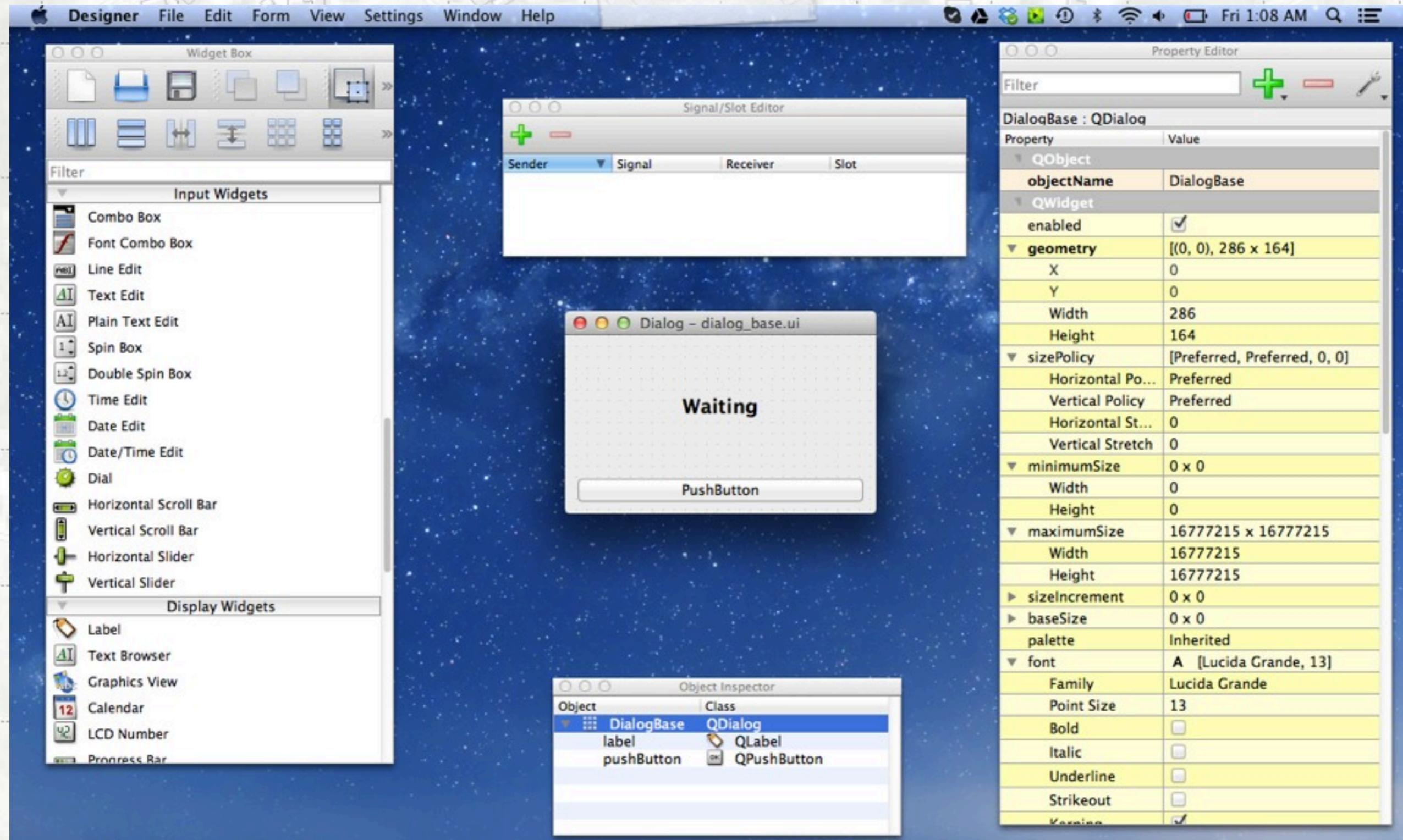
class HelloApp(Qt.QApplication):
    def __init__(self, args):
        Qt.QApplication.__init__(self, args)
        self.layout = QtGui.QVBoxLayout(self.widget)
        self.widget = QtGui.QWidget(None)
        self.button = QtGui.QPushButton("Click me", self.widget)
        self.label = QtGui.QLabel('Waiting...', self.widget)
        self.widget.setLayout(self.layout)
        self.layout.addWidget(self.label)
        self.layout.addWidget(self.button)
        # Call our slot (callback) whenever the button is pressed.
        self.connect(self.button, Qt.SIGNAL("clicked()"), self.slot)
        self.widget.show()

    def slot(self):
        self.label.setText('Hello, World!')

if __name__ == "__main__":
    app = HelloApp(sys.argv)
    app.exec_()
```



# Qt4 Designer 1



# Qt4 Designer 2

```
$ pyuic4 -o dialog_base.py dialog_base.ui  
$ ls  
dialog_base.py dialog_base.ui
```

Convert user interface (ui) file to python

# PyQt4 Application

```
#!/usr/bin/env python
# coding=utf-8
"""Example Qt4 Application"""

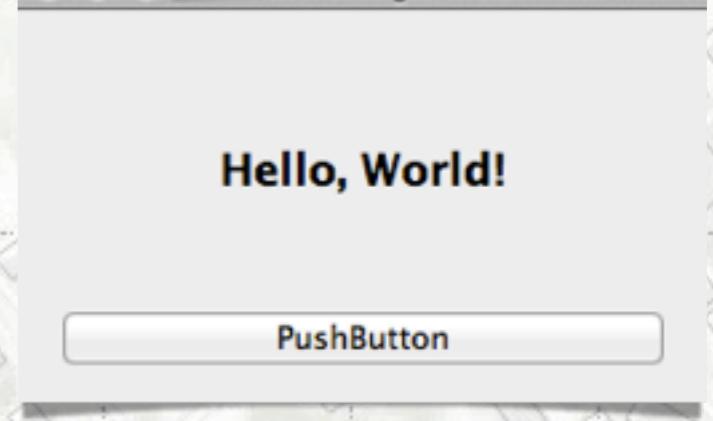
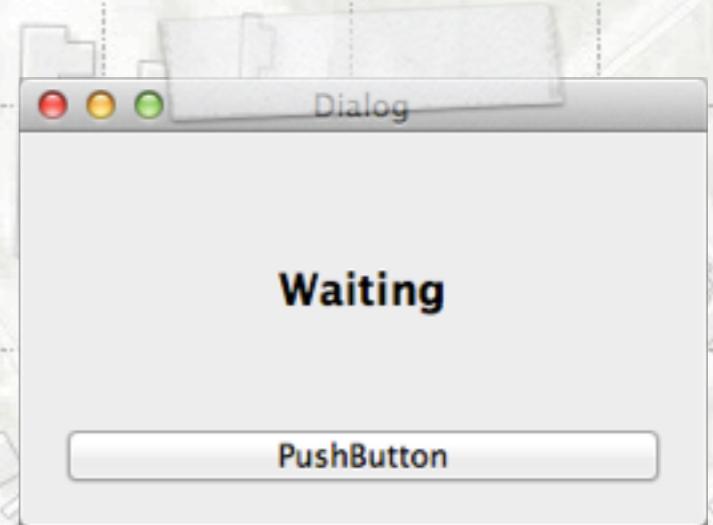
import sys
from PyQt4 import Qt, QtGui
from dialog_base import Ui_DialogBase


class Dialog(QtGui.QWidget, Ui_DialogBase):

    def __init__(self):
        QtGui.QWidget.__init__(self)
        self.setupUi(self)
        self.show()

    def on_pushButton_clicked(self):
        """Wow - an autoconnected slot!"""
        self.label.setText('Hello, World!')

if __name__ == "__main__":
    app = Qt.QApplication(sys.argv)
    dialog = Dialog()
    app.exec_()
```





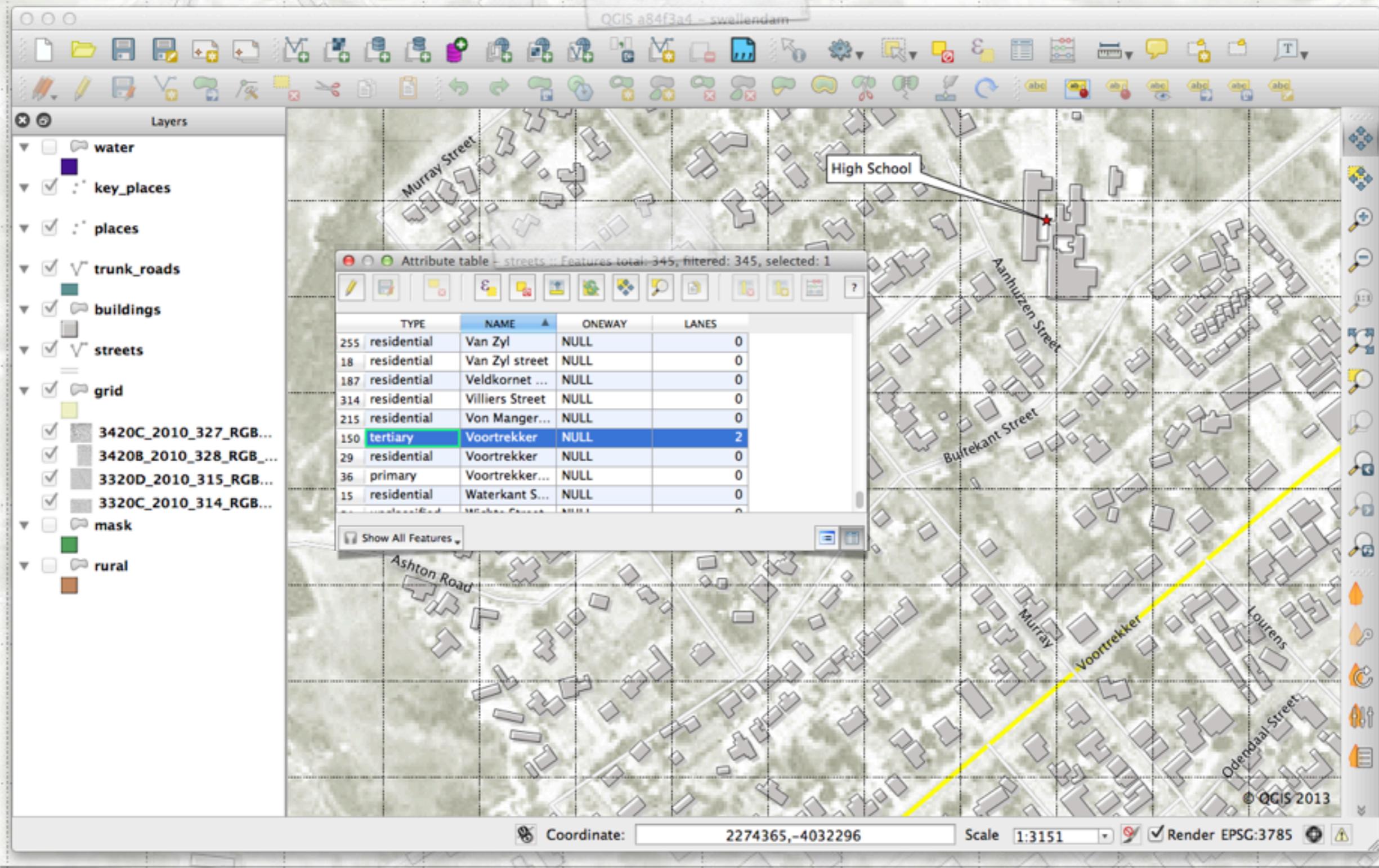
# Part 2

# QGIS

# What is QGIS?

- \* A free and open Geographical Information System
- \* Allows you to open, create, visualise and analyse geospatial data
- \* Completely Free and Open Source Software
- \* 11 Years old, under very active development

# QGIS Screenshot



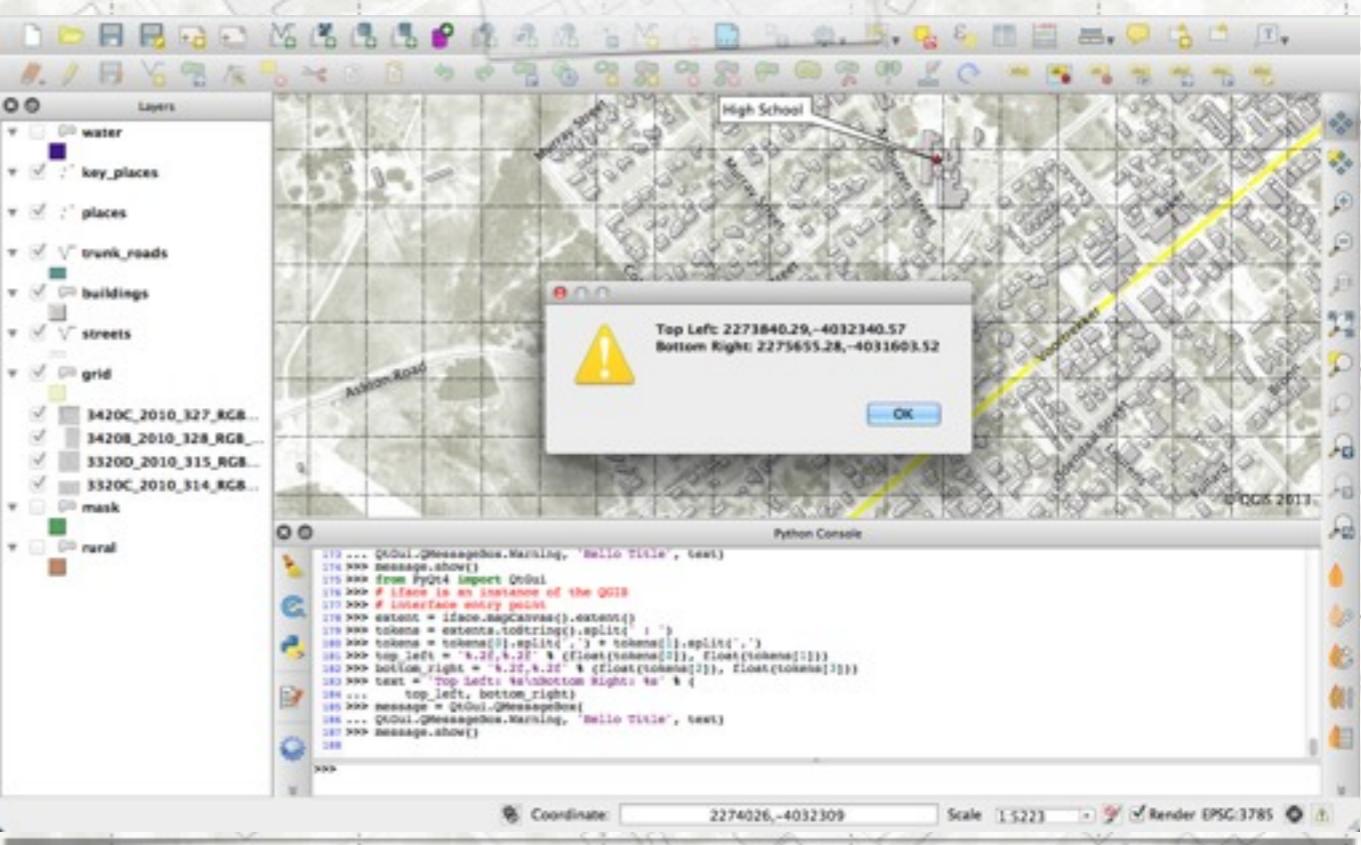
# QGIS Quick Tour

# QGIS 'Under the hood'

- \* Built with Qt4, PyQt4 written in C++
- \* QGIS-CORE - non-graphical libraries for working with GeoSpatial
- \* QGIS-GUI - re-usable GUI dependent elements (widgets, dialogs etc.)
- \* QGIS Application - the desktop application
- \* Python bindings for pythonistas!

# QGIS Hello World

```
# Note: This example created using QGIS master / 1.9
#
# from PyQt4 import QtGui
# iface is an instance of the QGIS
# interface entry point
extent = iface.mapCanvas().extent()
tokens = extent.toString().split(' : ')
tokens = tokens[0].split(',') + tokens[1].split(',')
top_left = '%.2f,%.2f' % (
    float(tokens[0]), float(tokens[1]))
bottom_right = '%.2f,%.2f' % (
    float(tokens[2]), float(tokens[3]))
text = 'Top Left: %s\nBottom Right: %s' % (
    top_left, bottom_right)
message = QtGui.QMessageBox(
    QtGui.QMessageBox.Warning, 'Hello Title', text)
message.show()
```

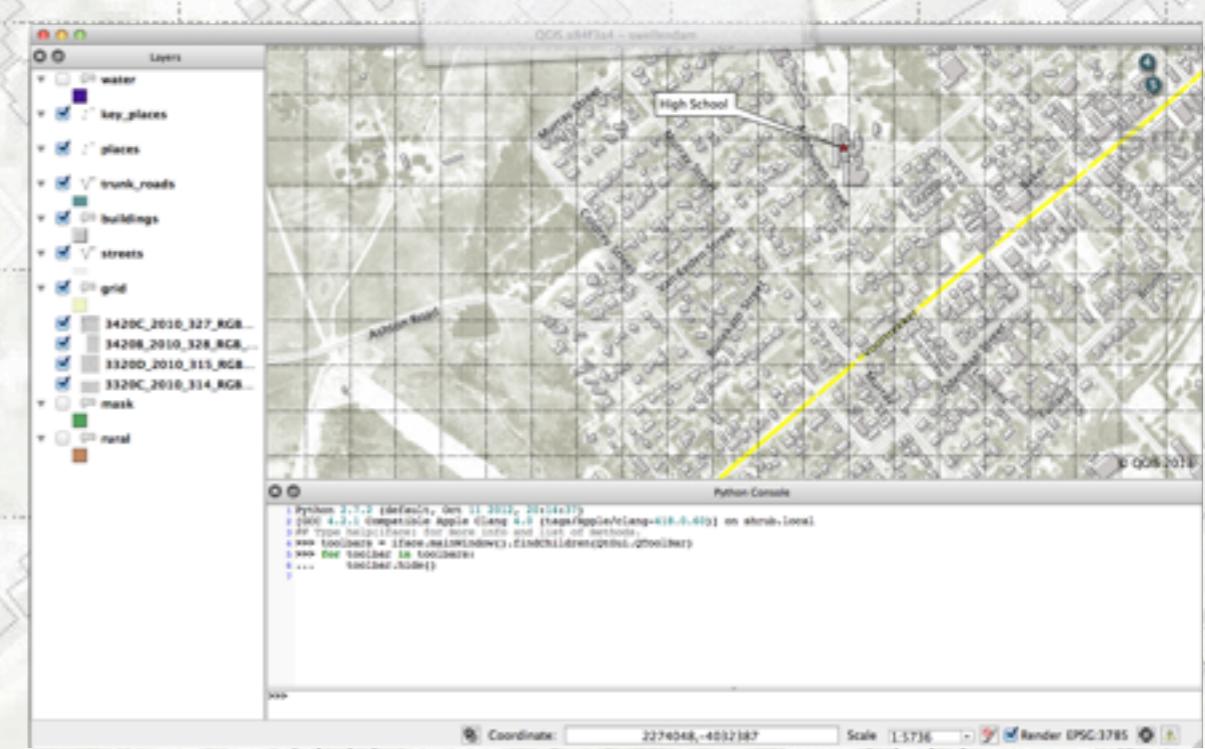
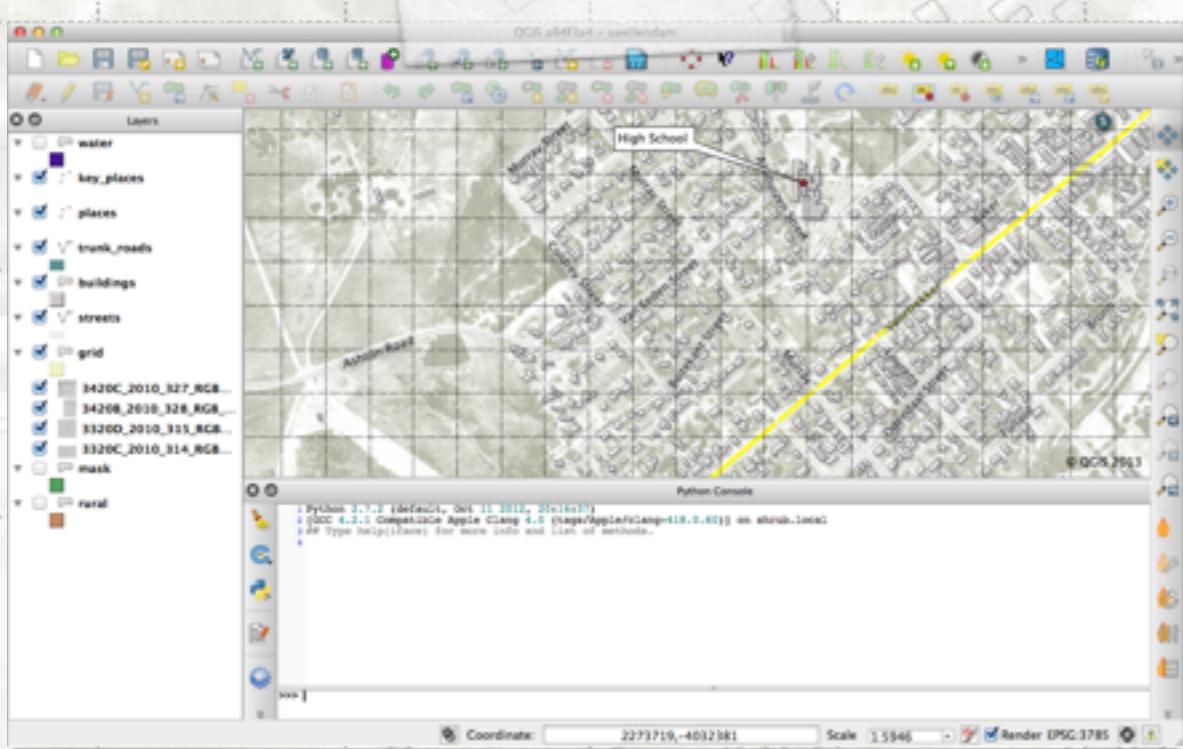


# QGIS API - What you can do...

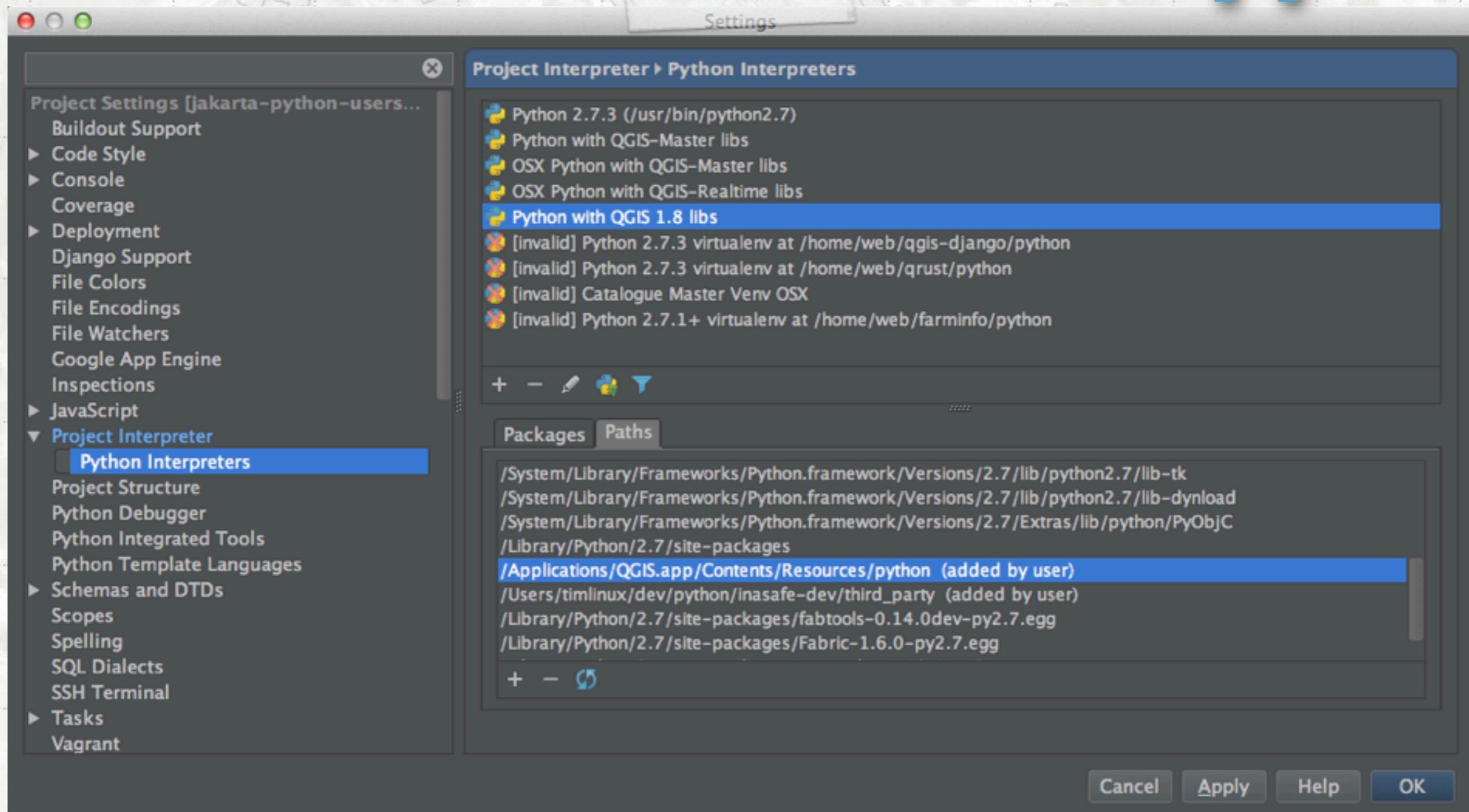
- \* Change at runtime almost any part of the application
- \* Create standalone GUI or server-side applications
- \* Create plugins that extend the functionality of QGIS

# Change almost any part of the app at runtime!

```
toolbars = iface mainWindow().findChildren(QtGui.QToolBar)
for toolbar in toolbars:
    toolbar.hide()
```



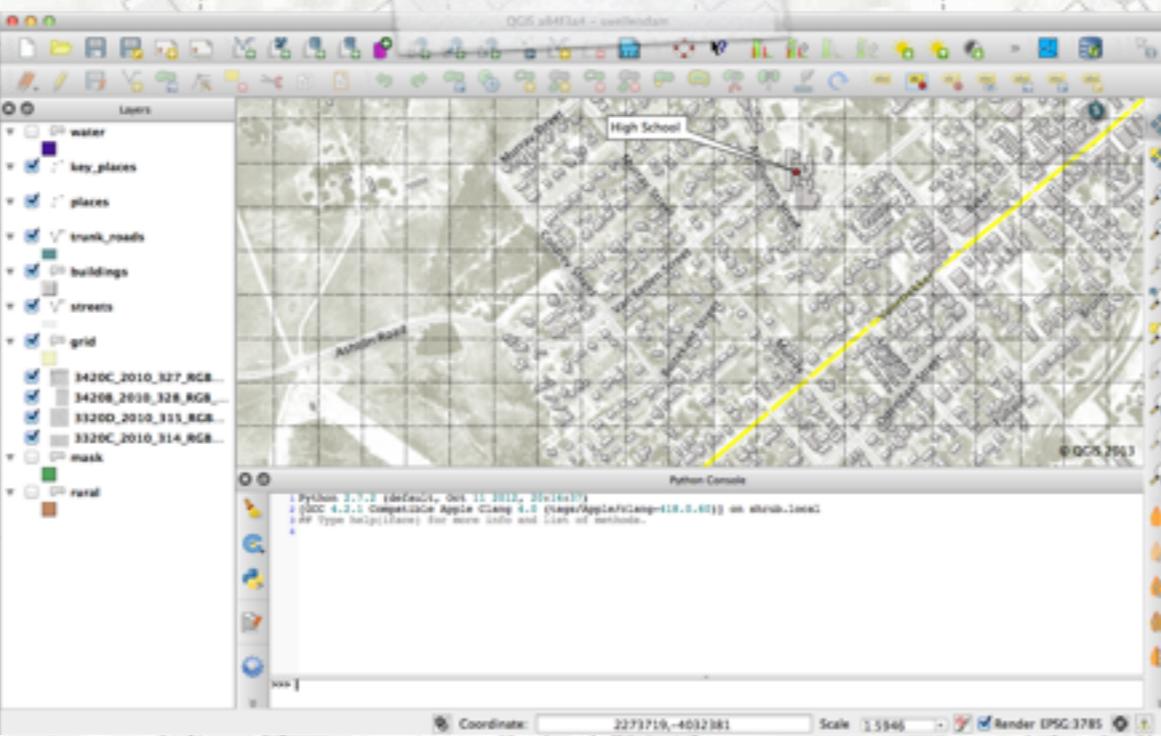
# Create standalone apps



Ensure that QGIS python directory is in your PYTHONPATH!

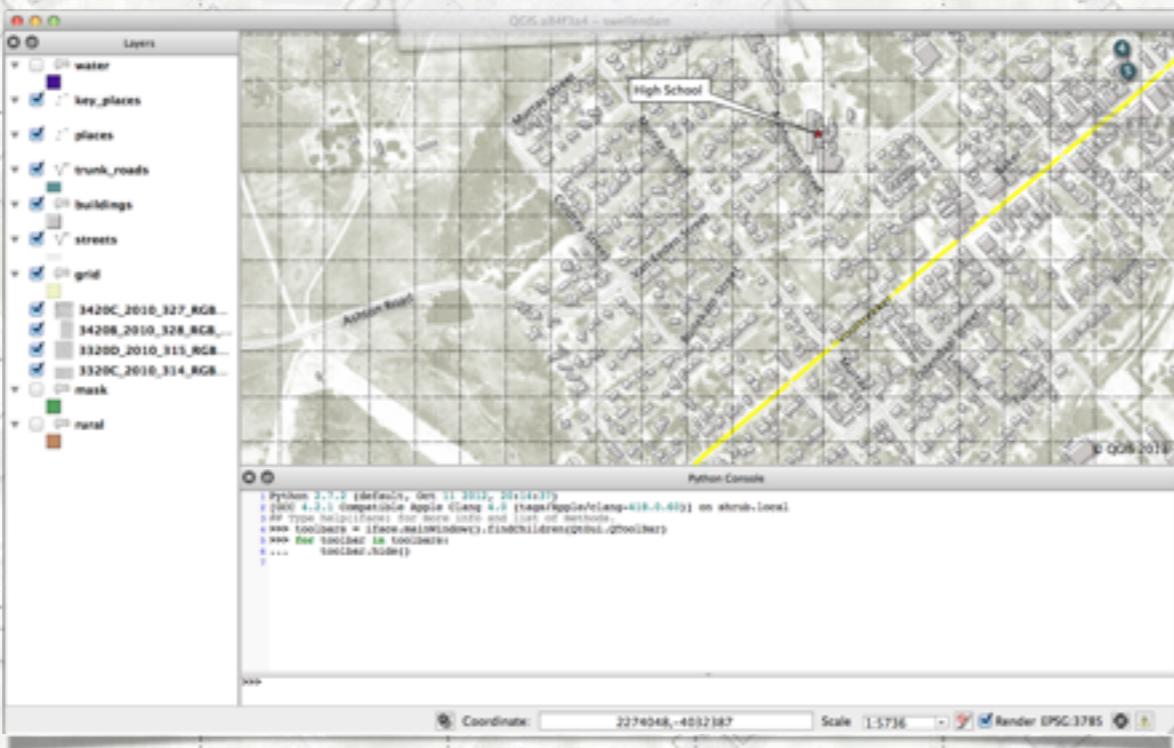
# Create standalone apps

Example of standalone app here....



# Create server side apps

Example of loading a layer and taking a screenie here...



# Part 3

# InaSAFE

# What is InaSAFE?

- \* A disaster contingency planning tool
- \* Supported by **AUSAID** and **BNPB** through the **Australia-Indonesia Facility for Disaster Reduction & World Bank's GFDRR**
- \* Completely Free and Open Source Software
- \* A library and a plugin for **QGIS**
- \* Written in **Python** (**numpy**, **PyQt4**)



**BNPB**



**THE WORLD BANK**

<http://inasafe.org>

# What is InaSAFE?

- \* BlackDuck 'Rookie of the Year 2012' Winner
- \* <http://www.blackducksoftware.com/open-source-rookies>



<http://inasafe.org>

# What is InaSAFE?

- \* Written by developers from around the world
- \* Hosted on GitHub <http://github.com/AIFDR/inasafe>



<http://inasafe.org>

# Demo of InaSAFE

# Part 4

# 10 Mantras for a busy Python Developer

# Mantra 1: Adhere to the element of least surprise

```
"""Mantra 1: Adhere to the principle of least surprise."""
```

```
def get_minimum(a, b):
    """Get the minimum value and cast to int..

    :param a: First of the numbers to compare.
    :param b: Second of the numbers to compare.
    :returns: The maximum value.
    :rtype: int
    """

    if a < b:
        return int(a)
    else:
        return int(b)

def get_maximum(a, b):
    """Get the maximum value and cast to int.

    :param a: First of the numbers to compare.
    :param b: Second of the numbers to compare.
    :returns: The maximum value.
    :rtype: int
    """

    if a > b:
        return a
    else:
        return b
```

# Mantra 2: Document, test, then code.

```
# coding=utf-8
"""Mantra 2: Document, test, code."""

def get_minimum(a, b):
    """Get the minimum value and cast to int..

    :param a: First of the numbers to compare.
    :param b: Second of the numbers to compare.
    :returns: The minimum value.
    :rtype: int
"""

pass

from unittest import TestCase

class TestMinimum(TestCase):
    """Tests for minimum."""
    def test_get_minimum(self):
        """Check if half returns the correct
value."""
        x = 10
        y = 55
        self.assertEqual(get_minimum(x, y), x)
```

# Now implement get\_minimum

# Mantra 3:

If a function is too long to fit on the screen, it is too long...

```
def too_long():
    """A very long function.

    :returns: An number representing the distance across the universe.
    :rtype: int
    """

if os.name != 'posix':
    # Poor guy is using windows
    string = (
        'We notice you are using windows. Oh shame what a pity to '
        'see you don\'t have unix underlying your system. That means '
        'no sed, grep, awk, cut, paste and all the other good things '
        'Gnu/Linux and OSX users enjoy...'
        'Unfortunately we are not able to calculate the distance accross '
        'the universe on a windows machine...'
        'If you are using Windows 3.1, please insert a floppy disk with '
        'your math card driver before we can continue. If you are using '
        'Windows 8, please look confused and spend some time wishing you '
        'were using windows 3.1 before you continue.'
        'If you are using Windows XP, please run up the hill with green '
        'grass that is displayed on your desktop.'
    )
    print string
    return

else:
    # Lucky guy has a unix like OS
    string = (
        'Ah - I see you have a unix like OS - congratulations on '
        'choosing well my friend. Proceeding to calculate distance across '
        'the universe.')
    print string

distance = 242342 # light years
```

```
def windows_message():
    """Poor guy is using windows."""
    string = (
        'We notice you are using windows. Oh shame what a pity to '
        'see you don\'t have unix underlying your system. That means '
        'no sed, grep, awk, cut, paste and all the other good things '
        'Gnu/Linux and OSX users enjoy...'
        'Unfortunately we are not able to calculate the distance accross '
        'the universe on a windows machine...')
    print string
    return
```

```
def unix_message():
    """Lucky guy has a unix like OS."""
    string = (
        'Ah - I see you have a unix like OS - congratulations on '
        'choosing well my friend. Proceeding to calculate distance across '
        'the universe.')
    print string
```

```
def just_fine():
    """A much more manageable function.

    :returns: An number representing the distance across the universe.
    :rtype: int
    """

    if os.name != 'posix':
        return windows_message()
    else:
        unix message()
```

# Mantra 4: Refactor continuously.

# Mantra 5: Don't self obfuscate your code.

```
# coding=utf-8
"""Mantra 5: Don't self obfuscate your code."""

# Taken from http://p-nand-q.com/python/obfuscated_python.html

# No normal human will ever make sense of this

fibonacci = lambda x:map(lambda o:(map(lambda c:map(lambda l:
o.__setslice__(l[0],l[1],l[2]),([o[2]+3,o[2]+4,[o[0]]],[0,3,[o[1]],
reduce(lambda x,o:x+o,o[:2]),o[2]+1]])),range(x)),o)[1],[[1,1,0]+
range(x)][0][3:]

print fibonacci(20)

def obfuscated():
    """Some hard to read code."""
    crn_ln = 100
    hgt = 200
    rslt = crn_ln + hgt
    return rslt

def clear():
    """Some easy to read code."""
    corner_length = 100
    height = 200
    result = corner_length + height
    return result
```

# Mantra 6: A coding standard

‘It doesn’t matter\* what your coding standard is, as long as you have one!‘

# Mantra 7:

Make it work first,  
then  
make it work fast

```
from timeit import Timer

def slow(a, b, c):
    """Add up three numbers.

    :param a: First number to add.
    :type a: int, float
    :param b: Second number to add.
    :type b: int, float
    :param c: Third number to add.
    :type c: int, float
    :returns: A number representing the sum of the three input numbers.
    :rtype : int, float
    """

    return sum([a, b, c])

def fast(a, b, c):
    """Add up three numbers.

    :param a: First number to add.
    :type a: int, float
    :param b: Second number to add.
    :type b: int, float
    :param c: Third number to add.
    :type c: int, float
    :returns: A number representing the sum of the three input numbers.
    :rtype : int, float
    """

    return a + b + c

if __name__ == "__main__":
    timer = Timer('slow(10, 20, 30)', setup="from __main__ import slow")
    print 'Slow:', timer.timeit()
    timer = Timer('fast(10, 20, 30)', setup="from __main__ import fast")
    print 'Fast', timer.timeit()
```

# Mantra 8: Make testing painless

[jenkins.linfiniti.com/job/InaSAFE-Release-Branch-QGIS1/](#)

# Jenkins

InaSAFE-Release-Branch-QGIS1

ENABLE AUTO REFRESH

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[GitHub](#)

[Coverage Report](#)

[Violations](#)

[SLOCCount](#)

[Git Polling Log](#)

[Build History](#) ([trend](#))

- #88 May 13, 2013 9:27:08 AM
- #87 May 13, 2013 8:39:41 AM
- #86 May 10, 2013 2:44:41 PM
- #85 May 8, 2013 1:45:41 PM
- #84 May 7, 2013 10:52:40 AM
- #83 May 6, 2013 5:21:06 PM
- #82 May 6, 2013 4:42:40 PM
- #81 May 3, 2013 9:07:41 AM
- #80 Apr 30, 2013 9:04:43 AM
- #79 Apr 17, 2013 11:10:43 AM

**Project InaSAFE-Release-Branch-QGIS1**

Builder for release branch of InaSAFE against QGIS 1.x

[edit description](#)

[Disable Project](#)

### Coverage Report

Recent Changes

Latest Test Result (no failures)

### Test Result Trend

count

Test Result Trend

(just show failures) [enlarge](#)

### Code Coverage

Classes	100% Conditions	100% Files	100% Lines	84% Packages
100%	100%	100%	100%	84%

%

Classes  
Conditionals  
Files  
Lines  
Packages

pep8 0 pylint 0

### Permalinks

- Last build (#88), 1 mo 16 days ago
- Last stable build (#88), 1 mo 16 days ago
- Last successful build (#88), 1 mo 16 days ago
- Last failed build (#43), 4 mo 22 days ago
- Last unstable build (#87), 1 mo 16 days ago
- Last unsuccessful build (#87), 1 mo 16 days ago

# Mantra 9:

If your code is not  
under version control  
it doesn't exist

Mantra 10:  
Python comes  
packed with great  
tools, use them!

pep8

pylint

Sphinx

# Conclusion

---

If you are a python programmer looking to do geospatial analysis and build GIS enabled applications, you are in luck! The tools I have shown you have everything you need to get started, and its all open source.



**B N P B**



**THE WORLD BANK**

# Thanks

My work on InaSAFE is supported by AUSAID and BNBP through the Australia-Indonesia Facility for Disaster Reduction.

The World Bank's GFDRR has also funded parts of InaSAFE and the addition of new features to QGIS that make our work possible.

[tim@linfiniti.com](mailto:tim@linfiniti.com)