# The path to developer happiness: Unit testing in Python & XSLT

When David Flanders asked me what I'd like to talk about at the Developer Happiness Days, I thought I could do something on XSLT – which turns out to be not that much of a hot topic – I ended up giving a short talk on XSLT and where to use it and where to not use it, and contributing to the Python day. In both cases I pushed the idea that beginners should start with test-first unit testing.

In the Python day we had a discussion about Integrated Development Environments and concluded that there is no one way, but Eclipse does seem to be popular (Ron Ward and I loathe it but others in our team use it happily). Someone pointed out to me that Eclipse is really good for multi-language development.

I did a quick demo of Doctest which is a way of embedding unit tests in documentation.

What's a unit test?

> In computer programming, **unit testing** is a software design and development method where the programmer verifies that individual units of source code are working properly. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class.
>
> http://en.wikipedia.org/wiki/Unit_testing

My contribution was a hello world program where the testable bit was a 'hi' function.

(And we didn't really say this but everyone I spoke to was in favour of developing with Unix based OS, meaning OS X or Linux (Ubuntu is the default now). If you're stuck with Windows then get some virtualization going.)

And here are my notes for the lighting talk on XSLT:

## XSLT? What is it? Why would you use it?

In this talk I'd like to cover the basics of:

- What it is?
- When to use it and when not.
- If you are going to use it, how to do so without losing your mind.

## What is XSLT?

1. A way of transforming XML into (almost always) another kind of XML.
2. XSLT is a plot by the computer science intelligentsia to sneak Lisp into your life.
3. XSLT is aggressively 'functionally' oriented (ie it's Lisp with an even scarier syntax).
4. XSLT is a standard widely available, simple way to transform data from one format to another.
5. XSLT 2 is an improvement but there is a severe shortage of implementations

Places I have used XSLT follow.

## The all-encompassing XSLT CMS

An entire web framework for content management. Site structure was defined by a 'site map' which was compiled into a big XSLT stylesheet used to process every request.

Do it again?

Emphatically no.

## Rendering word processing documents into HTML

Gets very complicated when trying to create hierarchical structure from flat input.

Modern word processing formats have too much indirection and too much application baggage to process with XSLT.

Do it again? Not until XSLT 3 :-)

## Simple data transforms

Simple data transforms for mapping data from one relatively flat form to another – eg EndNote references to Dublin Core metdata.

Works well enough.

Will do it again.

## If you're going to do it

Test first!

XSLT can really mess with your mind. Start with test cases. We use UTF-X.

http://utf-x.sourceforge.net/

And here's a quick how to get started on Ubuntu that I worked out in my prep for dev8d.

Summary:

6. Don't overdo it.
7. Use it where the overall structure of you input and output is either very congruent or very simple.
8. Write tests first.