Text Editors

P. Chambers, R. Entwistle

Next Generation Computational Modelling CDT
University of Southampton

FEEG6003 Advanced Computational Methods II

Outline

Preface

- Ensure you have downloaded the `.ova` appliance from `http://www.southampton.ac.uk/~ngcmbits/`
- When prompted, follow the exercises on the blog post: `http://computationalmodelling.bitbucket.org/tools/sublime.html`

Outline

Integrated Development Environments

(IDE's)

- Program development tool, typically including:
  - Graphical User Interface (GUI)
  - Source code editor
  - Automated build and run procedures
  - Output display
  - Debugging options
  - Variable Explorers
- Examples:
  - Spyder (Python)
  - Quincy (C)
  - TeXmaker (LaTeX) ...
- Many tools, user friendly
- Usually only work for one (or few) Computing Languages

UNIVERSITY OF Southampton

Integrated Development Environments

(IDE's)

- Program development tool, typically including:
  - Graphical User Interface (GUI)
  - Source code editor
  - Automated build and run procedures
  - Output display
  - Debugging options
  - Variable Explorers
- Examples:
  - Spyder (Python)
  - Quincy (C)
  - TeXmaker (LaTeX) ...
- Many tools, user friendly
- Usually only work for one (or few) Computing Languages

UNIVERSITY OF Southampton

Text Editors

- Used to edit text/code (intuitive)
  - Bare functionality ‘out of the box’
  - Fast operation through keystrokes
  - Can be customisable
  - Often cross-platform
  - Users need learn only one editor
- Open source:
  - Emacs, Vim, Nano ...
- Commercial:
  - Sublime Text[1], UltraEdit, Textmate...

[1] Sublime Text offers an untimed free trial

Text Editors

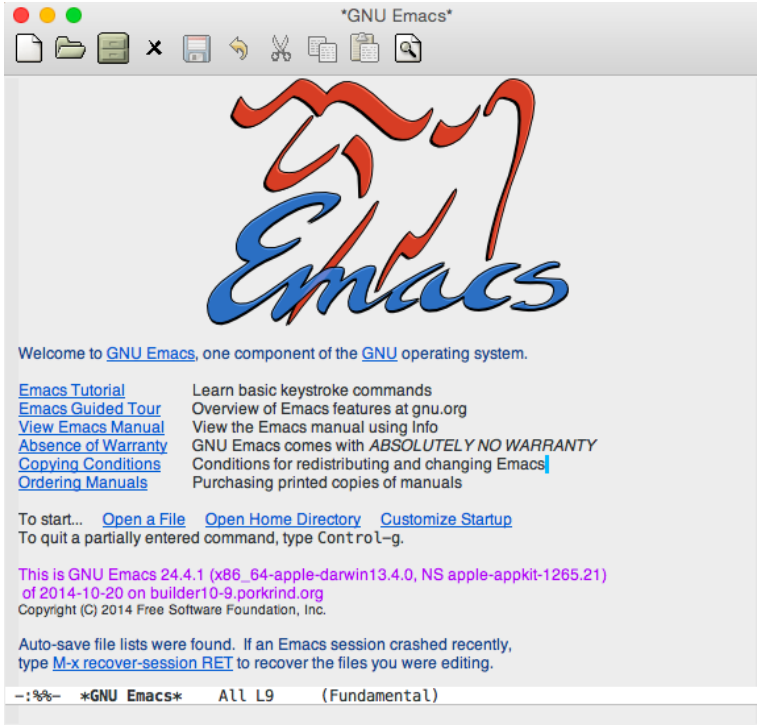
- Used to edit text/code (intuitive)
  - Bare functionality ‘out of the box’
  - Fast operation through keystrokes
  - Can be customisable
  - Often cross-platform
  - Users need learn only one editor
- Open source:
  - Emacs, Vim, Nano ...
- Commercial:
  - Sublime Text[1], UltraEdit, Textmate...

[1] Sublime Text offers an untimed free trial

Outline

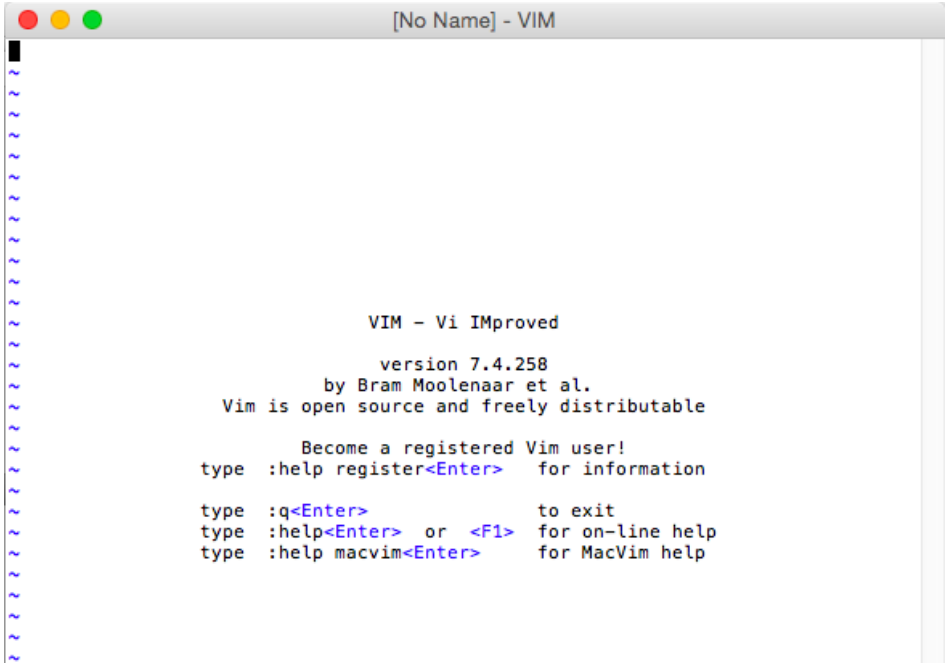
Emacs

- Free. GNU General Public Licensed, ‘GPL’
- Entirely cross-platform
- Highly customisable via Lisp language and package control
- Installed on most Linux Servers
- Steep (!) learning curve



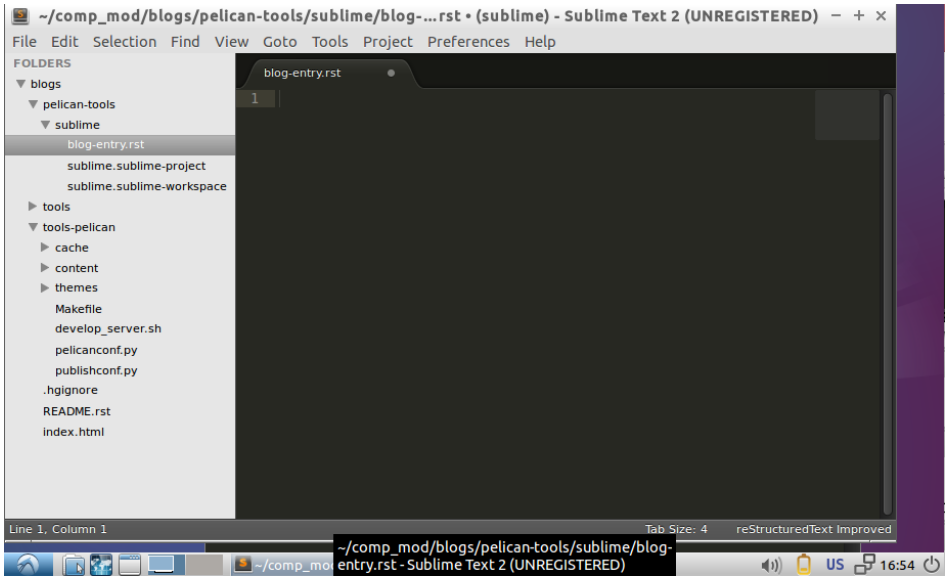
Vim

- Free. Open Source ‘Charityware’ - ‘GPL’ compatible
- Entirely cross-platform
- Customisable
- Based on Vi (default UNIX editor)
- Also a steep learning curve



Sublime

- Commercial user licensed - untimed free trial
- Windows, OSX, Linux
- Customisable via JSON settings files (simple!)
- Written in Python
- GUI only
- Not currently available on most servers



Outline

Sublime Features: General

- Multiple document tabs
- Side bar (Projects)
- ‘GoTo anything’
- Smart auto-complete
- Snippet expansion and suggestions
- Build Systems
- Keybindings and macros - Efficient
- Simple & instant customisation
  - Extensive package control
  - Plugins written in Python
  - JSON user settings - "attribute": "value"

Sublime Features: Projects

- Add/remove directories to the workspace: ‘Project’ tab
- Recover workspace in another session

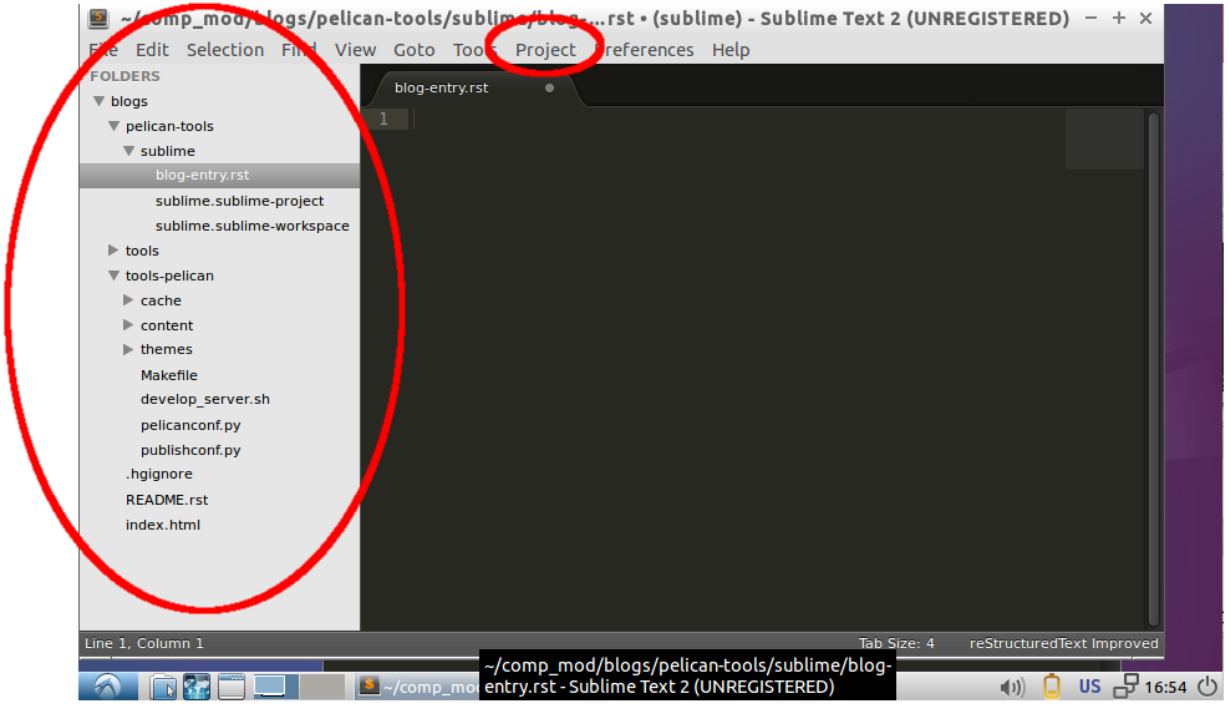


Figure: Projects in Sublime

Sublime Features: GoTo Anything

- Search file names and content in the current project
  - # - Fuzzy search document
  - : - Go to line
  - @ - Go to definition

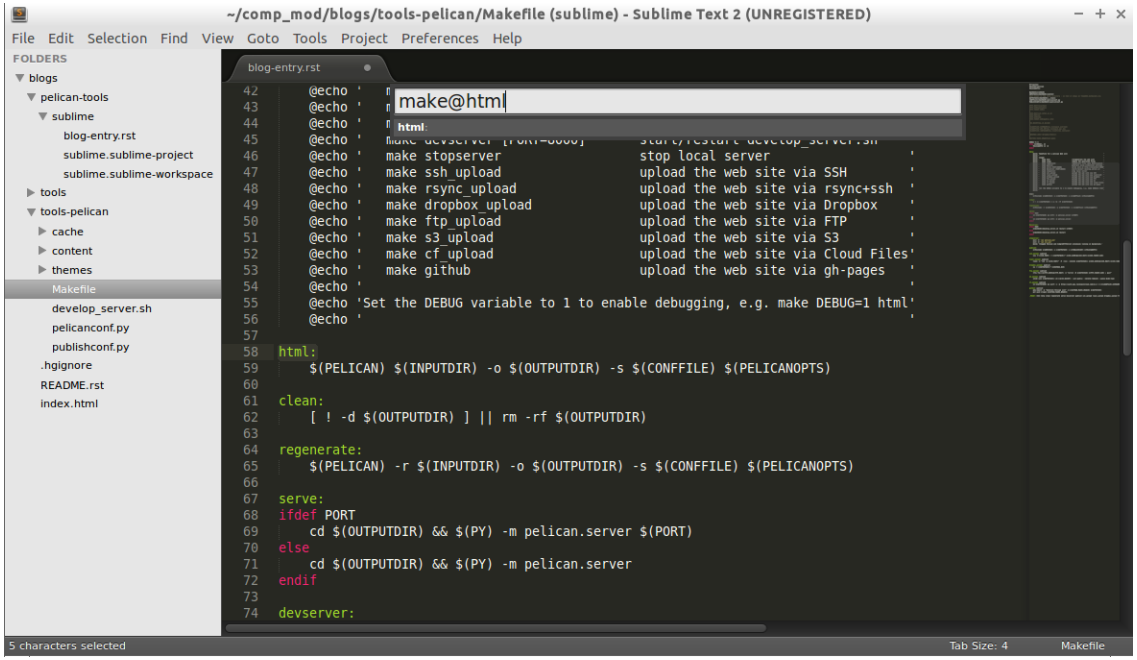
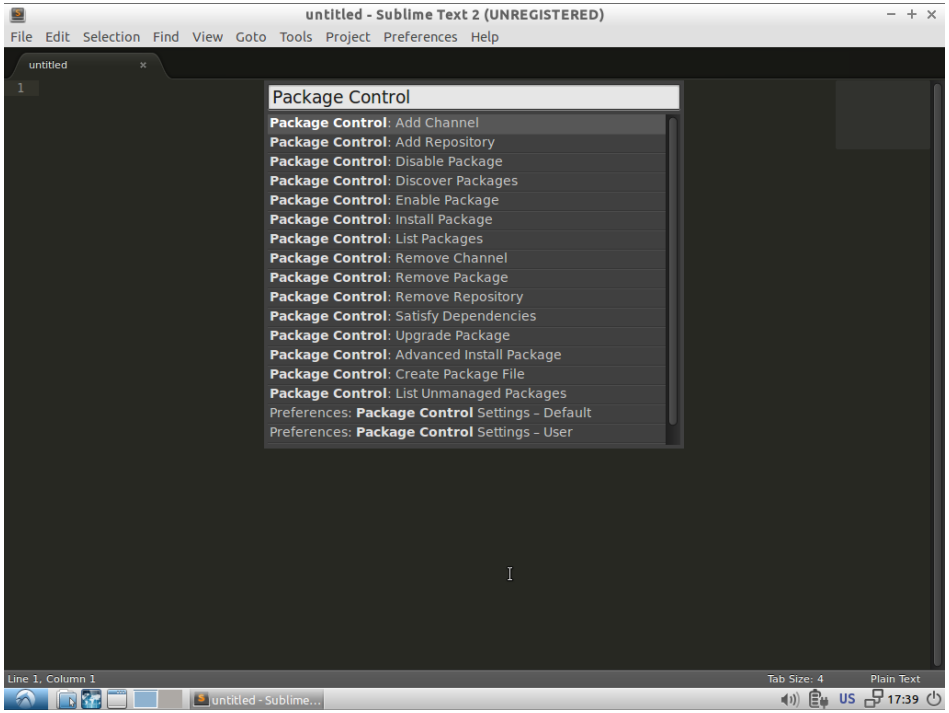


Figure: GoTo ‘html’ definition in Makefile in the current project

Sublime Features: Package Control

- Install packages via the ‘Command Palette’
- Packages available instantly (no restart needed)
- Automatic updates for installed packages



Targets for Practical (Part 1)

- Optimise ST2 for writing blog entries (reStructuredText)
  - Install ST2
  - Become familiar with the layout
  - Set up package control and install some packages

Blog Page:
`http://computationalmodelling.bitbucket.org/tools/sublime.html`

Outline

Snippets

- Smart templates
- Improve your efficiency

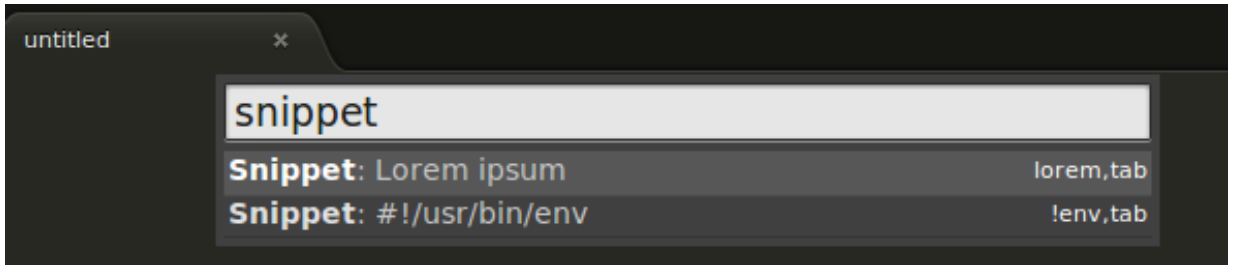


Figure: Using the Command Palette to find the list of snippets

New Snippet: Format

- *Tools > New Snippet...*
- XML file format
- Opens basic `Hello` snippet template

```
<snippet>
<content><![CDATA[
Hello, ${1:this} is a ${2:snippet}.
]]></content>
    <!-- Optional: Set a tabTrigger to define how to trigger the snippet -->
    <!-- <tabTrigger>hello</tabTrigger> -->
    <!-- Optional: Set a scope to limit where the snippet will trigger -->
    <!-- <scope>source.python</scope> -->
</snippet>
```

New Snippet: Tab key field markers

- Cycle through the fields you wish to update.

```
Hello, ${1:this} is a ${2:snippet}.
```

```
${#:abc}
```

- `${...}` - denotes a new field
- `1`, `2` - the cycle order
- `abc` - the text you will replace (use as a definition of the feild)

New Snippet: Pelican Blog Title

- Snippet: create blog title metadata quickly
- User defined snippets in *Packages* > *User*

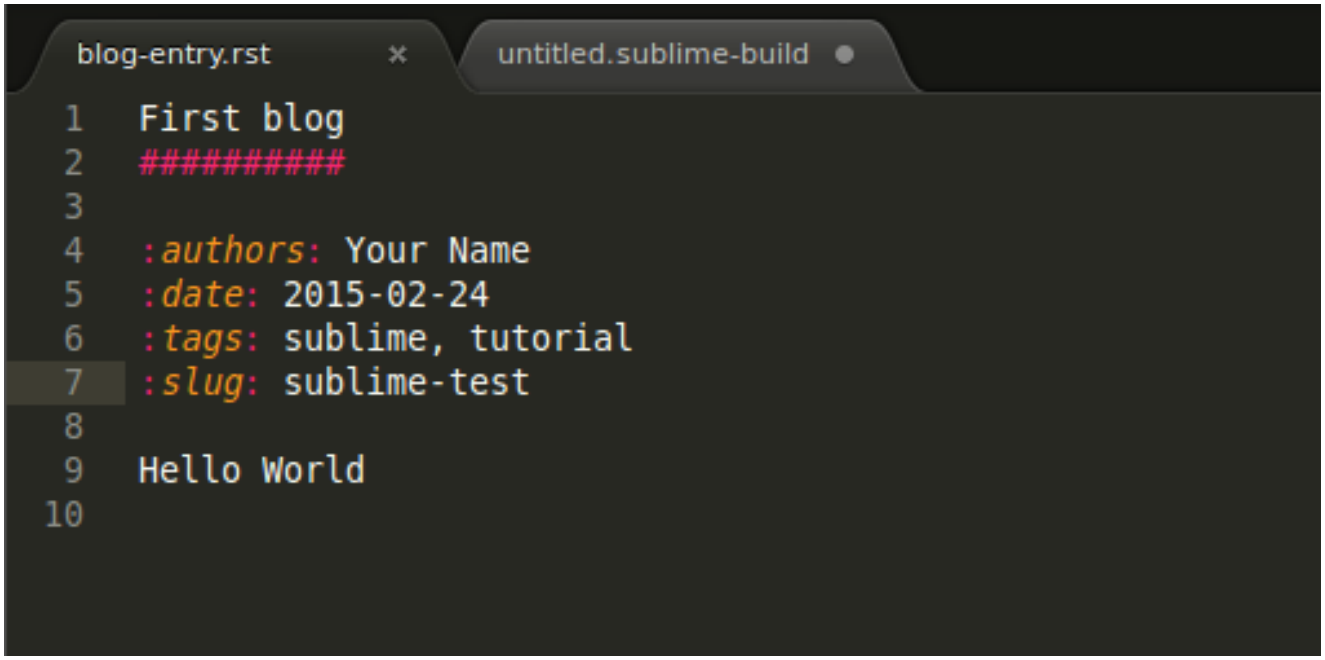


Figure: Blog entry title generated by the snippet

Outline

Build Systems

- Run files through external programs
- Displays the outputs

hello.c

```c
#include <stdio.h>

int main(void){
    /* Function that prints "Hello World" followed by a new line
    to the standard output and returns a interger */

    printf("Hello World\n");

    return 0;
}
```

```
Hello World
[Finished in 0.1s]
```

Line 11, Column 1 Tab Size: 4 C

Figure: Execution of a build system

New Build System: Format

- *Tools > Build System > New Build System*
- JSON file format
- Opens basic `make` JSON Build System

```
{
    "cmd": ["make"]
}
```

- See existing build-systems for more advanced settings

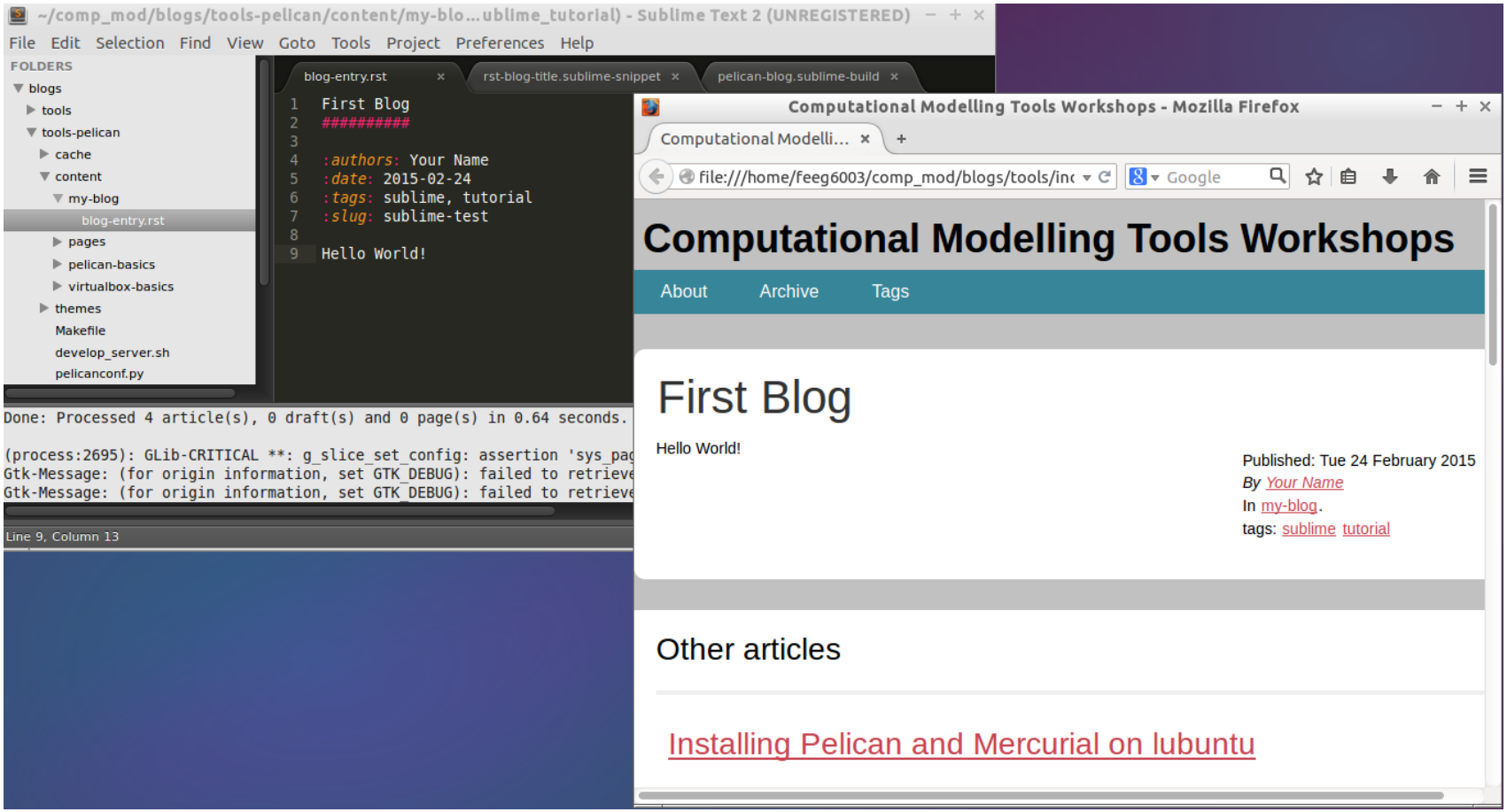
New Build System: Pelican

- Need Build System to:
  - Navigate to Makefile
  - Run `make html`
  - Open/view output

```
{
    "cmd": ["bash", "-c", "cd ~/comp-mod/blogs/pelican-tools && make html"],
    "selector": "text.restructuredtext"

    "variants":
    [
        {
            "name": "Run",
            "cmd": ["bash", "-c", "cd ~//comp-mod/blogs/pelican-tools &&
                  make html && cd ~/comp-mod/blogs/tools && firefox index.
                  html"]
        }
    ]
}
```

New Build System: Blog Preview



Targets for Practical (Part 2)

- Create a new snippet for blog title and metadata
- Create a build system for making the html page and viewing it locally

Blog Page:
`http://computationalmodelling.bitbucket.org/tools/sublime.html`

Summary

- ST2 is an intuitive text editor with many tools which work ‘out of the box’
- We have demonstrated ways to customise ST2 for writing Pelican blog entries
- Techniques presented should be transferable to other languages