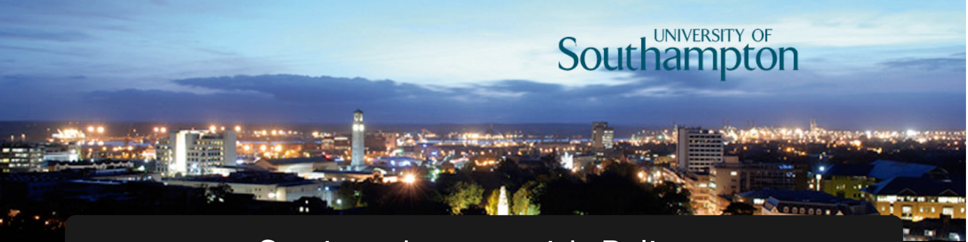


# Static webpages with Pelican

Denis Kramer

FEEG6003 Advanced Computational Modelling 2

12 February 2015

# Outline

UNIVERSITY OF Southampton

- Web technology basics
- Separation of content and presentation
- From content to webpage (Pelican)

---

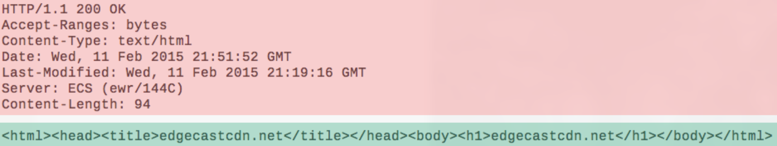
- Installing Pelican on lubuntu
- Cloning the blogs repository
- Start writing your topical blog

# HTTP request

1. Request
2. Response Header
3. Response Body

```
telnet www.example.com 80

GET /index.html HTTP/1.1
Host:  localhost
```



```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Type: text/html
Date: Wed, 11 Feb 2015 21:51:52 GMT
Last-Modified: Wed, 11 Feb 2015 21:19:16 GMT
Server: ECS (ewr/144C)
Content-Length: 94

<html><head><title>edgecastcdn.net</title></head><body><h1>edgecastcdn.net</h1></body></html>
```

# HTML

Hypertext-Markup Language is closely related to ⟨xml/⟩

```
This sentence has a
<a href="somewhere" style="color:red;">red link</a>.
```

1. `<tags>` and `</tags>` to mark regions of text
2. Attributes to add information for the browser
3. A `<html>` (or `<xml>`) document is a tree of tags and text

# Mixing content with presentation

```
This sentence has a
<a href="somewhere" style="color:red;">red link</a>.
```

1. href is content (it tells you where to go)
2. style is presentation (it tells the browser how to display)

# Cascading Style Sheets (CSS)

A web technology to move styling into separate document.

HTML file:

```
This sentence has a
<a href="somewhere">red link</a>.
```

CSS file:

```
a { color:  red; }
```

- Selectors to identify tags in the HTML source
- Style attributes to provide styling

# Extensible Stylesheet Language (XSL)

Allows to translate arbitrary XML documents into HTML

### XML document:

```
This sentence has a <link>red link</link>
```

### XSL definition:

```
<xsl:template match="link">
   <a style="color:red;"><xsl:apply-templates/></a>
</xsl:template>
```

### Transformed HTML:

```
This sentence has a
<a style="color:red;">red link</a>
```

# Roadmap

## Blog

http://computationalmodelling.bitbucket.org
/tools/pelican-basics.html

- Virtual Machine
  - Install VirtualBox (if not yet done)
    http://www.virtualbox.org
  - Download lubuntu image
  - Install Guest Extensions (optional)
- Install Pelican
  - Python 2.x
  - Make
  - Mercurial (for webpage repository)

# Installing the software stack

- Install Make and cpp

```
sudo apt-get install make gcc
```

- Install Guest Additions (optional)

```
cd /media/feeg6003/<NAME OF VBOXADDITIONS CD>
sudo ./VBoxLinuxAdditions.run
```

- Install Python package manager

```
cd $HOME
wget https://bootstrap.pypa.io/get-pip.py
sudo python get-pip.py
```

- Install Pelican

```
sudo pip install pelican markdown
```

- Install Mercurial

```
sudo apt-get install mercurial meld
```

# Clone blog repository

Blogs are hosted on `computationalmodelling.bitbucket.org`

## Clone repository

```
hg clone http://bitbucket.org/computationalmodelling/computationalmodelling.bitbucket.org blogs
```

*Hint:*
You can also fork the repository if you have a `bitbucket.com` account

# Repository structure

## Repository root

| | |
|---|---|
| tools | output directory |
| tools-pelican | source directory |
| readme.rst | Read-me file (read it!) |
| index.html | Dummy html file for host redirects |

## Source folder (tools-pelican)

| | |
|---|---|
| content | source folder with blog entries |
| Makefile | defines targets for make tool |
| *.py | Pelican configuration files |
| themes | folder for themes |
| develop_server.sh | starts a development web server locally |

# Start your own blog

- Go to content folder

```
cd content
```

- Create folder for your blog entry and go there

```
mkdir <name>
cd <name>
```

- Create a Markdown (reStructuredText) file for your blog

```
nano blog-name.md
```

# Blog entry

## Metadata

```
title:  My amazing first blog
authors:  You know who
date:  2015-02-12
tags:  demo, training
slug:  demo-post
```

## Link

```
I learned how to use Pelican for blogging
[here]({filename}/pelican-basics/pelican-basics.md)
```

# Translating to HTML

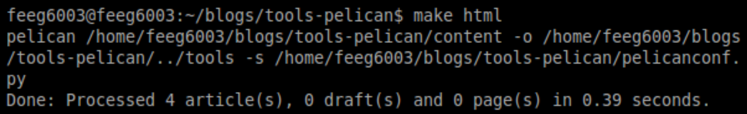
## Running Pelican

- Make sure you are in the *tools-pelican* folder where the *Makefile* resides

  ```
  cd ../..
  ```

- Call Pelican using the *make* utility with the *html* target

  ```
  make html
  ```



```
feeg6003@feeg6003:~/blogs/tools-pelican$ make html
pelican /home/feeg6003/blogs/tools-pelican/content -o /home/feeg6003/blogs
/tools-pelican/../tools -s /home/feeg6003/blogs/tools-pelican/pelicanconf.
py
Done: Processed 4 article(s), 0 draft(s) and 0 page(s) in 0.39 seconds.
```

# Testing the page locally

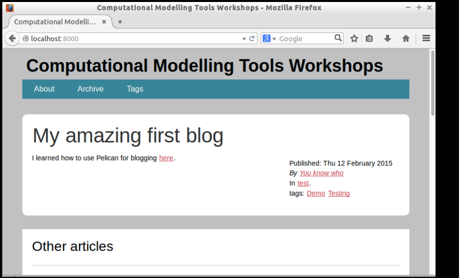
## Test in local web server

- Use *make* with target *serve* to start a lightweight webserver

  make serve

- Point web browser to local port 8000

  http://localhost:8000



Computational Modelling Tools Workshops

About Archive Tags

My amazing first blog

I learned how to use Pelican for blogging here.

Published: Thu 12 February 2015
*By* You know who
In test.
tags: Demo Testing

Other articles

Notice that the web server blocks the terminal while running. You can close the server with *Ctrl+C*.

Happy blogging!