

Setting up a Website/ Blog

using Static site generators

Conor Gilmer BSc
Dublin
Ireland

January 17, 2015

Contents

1	Static website generators	4
2	Jekyll	5
3	Hyde	6
3.1	Install	6
3.2	Create Hyde Project	6
3.3	Generate Site and Run on Local Server	6
4	Pelican	7
4.1	Installation	7
4.2	Adding Content	7
4.2.1	Pages	7
4.2.2	Blogs	7
4.3	Generating the Website	8
5	Deploying/Publishing	9
5.1	Github Pages	9
5.2	DropBox	9
5.3	FTP to a Host	9
6	Nikola	10
7	Markdown	11
7.1	Headings	11
7.2	Bold, Italics or Strikethrough	11
7.3	Links	11
7.4	Lists	12
7.4.1	Unordered Lists	12
7.4.2	Ordered Lists	12
7.5	tables	12
7.6	Inserting an Image	12

8	Conclusion	13
A	Appendix 1	15
A.1	sitemap.xml	15

1 Static website generators

Recently I have begun to look again at static website generators for creating websites and blogs. Static Website Generation is not a new concept, I have used a number of Perl based ones and even LaTeX in the past, but more recently they have seemed to get some energy as alternatives to a Content Management System (CMS) and particularly as they suit documentation and publishing to some cloud services such as *github pages*. There is a number of them about and I am going to outline my thoughts and how to use them.

- Jekyll
- Hyde
- Pelican
- Nikola

2 Jekyll

Is a Ruby Static website generator, it seems quite favoured with github pages No specific ruby coding skills are needed, but ruby is needed.

3 Hyde

Is a Static website generator which uses python, similar to its alter ego Jekyll. While it uses python, no specific python knowledge is required to use it. (In my case I used Python 2.7 installed with Anaconda) Hyde uses a Django based templating system, Jinja2.

3.1 Install

The easiest way to install Hyde was to use *easy_install* or *pip install*.

```
sudo easy_install hyde
or
pip install hyde
pip install jinja
```

3.2 Create Hyde Project

```
mkdir hydesite
cd hydesite
hyde -s create
```

3.3 Generate Site and Run on Local Server

You then generate the static html, and you can view it by running the hyde server.

```
hyde gen
hyde serve
```

4 Pelican

Is a Static website generator which uses python, although again no specific python skills are required to use the system.

4.1 Installation

```
pip install pelican
pip install Markdown
mkdir mysite
cd mysite
pelican-quickstart
```

This uses *pip* to install pelican and markdown, *pelican-quickstart* sets up the plumbing of your pelican website.

The site is configured using settings.py (or pelicanconf.py). The layout is set out in the theme which is used, at the moment we will use one of the simple themes notforme. You can change a theme or modify it or compose your own by editing the css and templates for the theme.

4.2 Adding Content

Here we will have two types of content on the website standard pages, and blog posts.

4.2.1 Pages

Pages are stored in the pages directory in the content directory e.g. about.md, contact.md etc. they are traditional webpages and are not listed or timestamped with author or category information as Blog post or news item is.

Title: About

```
##This is About page
This is a blog generated by Pelican using Python...
```

(if you were just using it to generate a static website without any news items or blog you would just have the pages of the site here.

4.2.2 Blogs

Any markdown page in the content directory which is in a directory other than pages is treated as a blog.

Date: 2015-1-9

Title: My First Blog Post

```
##this is my first blog post in pelican##  
Here is the main text of my first blog post
```

The Date is the time the post will timestamped at, and you can have minutes as well, the Title is the title you will see and the

4.3 Generating the Website

```
pelican content/ -s settings.py
```

The site is then generated in the *output* directory.

5 Deploying/Publishing

5.1 Github Pages

Most of the newer static website generators are conveniently setup for publishing on github pages (username.github.io).

5.2 DropBox

You can also deploy to your Dropbox account.

5.3 FTP to a Host

You can just FTP the contents of the output directory to you host, the website should be generated to the domain and page of your host.

```
git add .  
git commit -m "updates"  
git push origin master
```

6 Nikola

Another static website generator.

7 Markdown

Since i have used Markdown for the above here is a few items from it

7.1 Headings

Headings are indicated by the Hash Symbol, the number of which indicate if it is H1, H2, H3 etc.

```
#This is H1
##This is H2
###This is H3
####This is H4
#####This is H5
#####This is H6
```

7.2 Bold, Italics or Strikethrough

For empasis Bold or Italics are often used, and sometime text needs to have a line through it.

Italics uses **asterisks** or *_underscores_*resulting in *italicised*

Bold uses ****two asterisks**** or **__two underscores__**resulting in **bolded**

Strikerhough is achived by using two tildes(~) e.g. ~~2tildes~~ results a line through the text.

7.3 Links

A link can be as simple, where just the URL is written

```
<http://www.gnu.org>
```

A piece of text can be set as a link to a URL

```
[Pelican](http://blog.getpelican.com/)
```

You can add ALT text to it, in the case below Get Pelican would display when your mouse hovers over the link Pelican.

```
[Pelican](http://blog.getpelican.com/ "Get Pelican")
```

7.4 Lists

7.4.1 Unordered Lists

You can use +, - or * to an unordered list

- * First Item
- * Second Item
- * Third Item

- First Item
- Second Item
- Third Item

- + First Item
- + Second Item
- + Third Item

7.4.2 Ordered Lists

- 1 First Item
- 2 Second Item
- 3 Third Item

7.5 tables

A	B	Result	
-----	-----	:-----:	
True	True	True	
True	False	False	
False	True	False	
False	False	False	

7.6 Inserting an Image

Inserting an image with markdown.

```
![Pelican](../../images/pelican.jpg)
```

In this case i have an image in a directory in the main file so to be accessed from a markdown file in the /content/pages/ directory you have to specify the page

8 Conclusion

To me the benefits of using a static website generator, are speed of a static website, no time delay due to interaction with a database or dynamic page generation, great power to customise the site with editing the themes and editing the css. It does take the composition of content away from concerns of how it is displayed on the page. Also it does provide something different from the ubiquitous wordpress CMS websites which seem about. The negatives are that it does require some technical nous to set up, its graphical side may be limited to what is available and your own graphic knowhow, it doesn't have the dynamic functionality which plugins provide for many CMS systems. Overall I like SSG and I think they are a powerful tool to generate a website.

References

- [1] Uche Ogbuji, *Build rapid and lightweigh websites*, IBM Developer Works (February 2013), <http://www.ibm.com/developerworks/library/wa-hyde/wa-hyde-pdf.pdf>.

A Appendix 1

A.1 sitemap.xml

©2014, 2015 Conor Gilmer all rights UNreserved.