

# Jekyll + Liquid, Static Site Generation with Ruby

Jon Ander Hernández



EHUko Software Librea sustapenerako taldea

14 de Mayo de 2009



**1** Introduction

**2** Example

**3** Plugins

**4** References



## How did I discover Jekyll?

# GitHub

- ▶ A web-based hosting service for software development projects that use the Git revision control system.
- ▶ Provides social networking functionality such as feeds, followers, ...






## Linus Torvalds

torvalds

-  Linux Foundation
-  Portland, OR
-  Joined on Sep 03, 2011

**10k** followers   **2** starred   **0** following

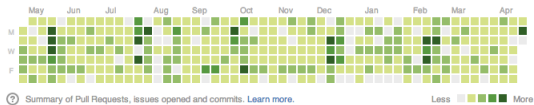
 Contributions    Repositories    Public Activity

 Follow   

### Popular repositories

-  **linux** 8,363 ★  
Linux kernel source tree
-  **subsurface** 398 ★  
Multi-platform develog in C and Gtk

### torvalds's Open Source Contributions



**2,318 Total**

Apr 30 2012 - Apr 30 2013

Year of Contributions

**29 days**

September 16 - October 14

Longest Streak

**4 days**

April 26 - April 29

Current Streak

### Contribution Activity

Period: **1 Week** ▾

 **33** Commits

**Pushed 33 commits to torvalds/linux** Apr 24 - Apr 29

PUBLIC  mojombo / jekyll Watch ▾ Star

10,629

 Fork

1,816



Code

Network

Pull Requests 23

Issues 50

Wiki

Graphs

All Requests

23


















Yours

maul-esel	5
benbalter	4
parkr	2
x3ro	1
sillylogger	1
dhcole	1
darwin	1
stephenmod	1
ixtl	1
cobyism	1
uu59	1
liufengyun	1
goodtouch	1
stomar	1
juryzicka	1

Open

Closed

Sort: Newest ▾

-  **Fix `jekyll serve --limit\_posts n` failed** #1004  
 Just fix this error: \$ ./bin/jekyll serve --limit\_posts 2 --trace Configuration file: /home/nyara...  
 by uu59 3 days ago  11 comments
-  **support data sources** #1003  
 Data Source enables you to load data from external sources, such as web, file system, databases a...  
 by liufengyun 3 days ago
-  **Use post's directory path when matching for the post\_url tag** #998  
 Adds support for matching file names of posts in subdirectories. Works with the following: {% pos...  
 by dhcole 5 days ago  13 comments
-  **Use valid css class names when pygments language contains invalid chars, eg html+django** #994  
 Pygments has some lexers that combine multiple languages and are referenced by name using the plu...  
 by stephenmod 7 days ago  4 comments
-  **add leading slash to page.url and post.url** #992  
 Liquid: {% for page in site.pages %} {{ page.url }} {% endfor %} Example output: /docs/api/deve...  
 by maul-esel 11 days ago  11 comments
-  **Move top, header, and footer includes into default layout.** #986  
 This is a design pattern / best practices decision and would love everyone's thoughts. Following ...  
 by benbalter 12 days ago  1 comment



itsos

PUBLIC  mojombo / jekyll Watch Star

10,629

 Fork

1,816



Code

Network

Pull Requests

23

Issues

50

Wiki

Graphs

Browse Issues

Milestones

Search:

Issues &amp; Milestones...



New Issue

Everyone's Issues

50

Created by you

0

Mentioning you

0

No milestone selected



## Labels



bug

8



discussion

3



docs

1



enhancement

5



feature

6



migrator

1



@mojombo

1



needs-decision

2



question

4



suggestion

6



chopping block

0

50 Open

952 Closed

Sort: Newest



Hard dependency on Pygments

#1006

Opened by völdnulf 17 hours ago



Fix `jekyll serve --limit\_posts n` failed

#1004

Opened by uu59 3 days ago  4 comments

support data sources


#1003

Opened by ßufengyun 3 days ago




Use post's directory path when matching for the post\_url tag

#998

Opened by dhcole 5 days ago  6 comments

1.0.0.rc1 failure: undefined method `xmismatch'

#997

Opened by jonforums 5 days ago  15 comments

Use valid css class names when pygments language contains invalid chars, eg html+django

#994

Opened by stephenmcd 7 days ago  4 comments

add leading slash to page.url and post.url

#992

Opened by maur-esel 11 days ago  5 comments

page.url returns different values between posts and pages

#991

Opened by xhan 11 days ago



itsos

# Github provides hosting for projects, users and repos

- ▶ Hosting

- ▶ Create a repository named *username.github.io* or *orgname.github.io*.

- ▶ Setting up a custom domain

- ▶ Create a file named CNAME in the root of your pages and put the domain (or subdomain) into the file.
- ▶ Create a CNAME DNS record: `www.example.com` → `username.github.io`.



# Jekyll is the engine behind GitHub Pages.

Go to the repo's settings page

## GitHub Pages

Create a beautiful site for your project with our [GitHub Pages](#) generator.

Author your content in our markdown editor, select a theme, then publish.

To publish a page manually, push an HTML or [jekyll](#) site to a branch named `gh-pages`. [More info](#).

Automatic Page Generator












## Project Name

## Tagline

Body (parsed with [GitHub Flavored Markdown](#))

**h1** **h2** **h3**   **B** */* **<>**     

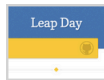
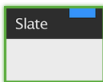
[Load README.md](#)

### Welcome to GitHub Pages.

This automatic page generator is the easiest way to create beautiful pages for all of your projects. Author your page content here using GitHub Flavored Markdown, select a template crafted by a designer, and publish. After your page is generated, you can check out the new branch:

```
...  
$ cd your_repo_root/repo_name  
$ git fetch origin  
$ git checkout gh-pages  
...
```





The screenshot shows a GitHub Pages page for a project named "Panchat". At the top, there are navigation buttons: "HIDE", "EDIT", and "PUBLISH". A blue button with the GitHub logo says "View on GitHub". The main heading is "Panchat" in a large, white font, followed by the subtitle "A simple chat with FIFO, Causal and Total order". Below this, there are two folder icons labeled "tar.gz" and ".zip" with a downward arrow pointing to them. The main content area has a light gray background and contains the text "Welcome to GitHub Pages." followed by a paragraph: "This automatic page generator is the easiest way to create beautiful pages for all of your projects. Author your page content here using GitHub Flavored Markdown, select a template crafted by a designer, and publish. After your page is generated, you can check out the new branch:". Below the text is a white box containing a code block with three lines of terminal commands.

```
$ cd your_repo_root/repo_name
$ git fetch origin
$ git checkout gh-pages
```

# Jekyll rules!

Ruby + YAML + Liquid = AWESOME!



- ▶ Ruby
- ▶ Liquid
- ▶ YAML
- ▶ Markdown
- ▶ Jekyll



# Ruby language



- ▶ Dynamic typing and duck typing
- ▶ Dynamic reflection and alteration of objects
- ▶ Centralized package management through RubyGems

```
gem install jekyll json mini_magick
```

- ▶ Implemented on all major platforms
- ▶ Large standard library



## How does it look like?

- ▶ Literal notation for arrays, hashes, regular expressions and symbols

```
hash = Hash.new

hash = { :water => 'wet', :fire => 'hot' }

puts hash[:fire] # Prints: hot
```

- ▶ Thoroughly object-oriented with inheritance, mixins and metaclasses

```
"ice is nice".length # 11
"ruby is cool.".index("u") # 1

class Person
  attr_reader :name, :age
  def initialize(name, age)
    @name, @age = name, age
  end
  def <=>(person) # Comparison operator
    age <=> person.age
  end
end

class Hacker < Person
end
```



- ▶ Lexical closures, iterators and generators, with a unique block syntax

```
(1..3).map { |x| x * 2 } #=> [2, 4, 6]
```

```
hash.each do |key, value|  
  puts "#{key} is #{value}"  
end
```

```
def create_set_and_get(closure_value=0)  
  return Proc.new {|x| closure_value = x},  
         Proc.new { closure_value }  
end
```

```
setter, getter = create_set_and_get  
setter.call(21)  
getter.call # => 21
```

# YAML, a human-readable data serialization format

```
customer:
  given: Dorothy
  family: Gale

# ["customer"]["given"] => "Dorothy"

items:
  - part_no: A4786
    descrip: Water Bucket (Filled)

  - part_no: E1628
    descrip: High Heeled "Ruby" Slippers

items2: [ { part_no: A4786 }, { part_no: E1628 } ]

# ["items"][0]["part_no"] => A4786

bill-to: &id001
  street: >
    Wrapped text
    will be folded
    into a single
    paragraph

refer: *id001
```



# Liquid, a template engine

- ▶ Non evaling and secure
- ▶ Stateless. Compile and render steps are separate.
- ▶ Just render passing in a hash with local variables and objects.

```
# Parses and compiles the template
@template = Liquid::Template.parse("hi {{name}}")

# => "hi tobi"
@template.render('name' => 'tobi')
```

- ▶ Markup composed of tags, blocks and filters.



```
<ul id="products">
  {% for product in products %}
    <li>
      <h2>{{ product.name }}</h2>
      Only {{ product.price | price }}

      {% if product.description %}
        {{ product.description | prettyprint | paragraph }}
      {% else %}
        No description
      {% endif %}
    </li>
  {% endfor %}
</ul>
```

Remember, everything is non evaling code, if, for, ...are simple tags/blocks.



# Markdown, a lightweight markup language

Used on GitHub, reddit and Stack Overflow

Write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML

```
# First-level heading
```

```
#### Fourth-level heading
```

1. An item in an enumerated (ordered) list
  - 1.1. A subitem, indented with 4 spaces
2. Another item in an enumerated list
5. Yet another item

```
*emphasis* or emphasis (e.g., italics)
```

```
**strong emphasis** or strong emphasis (e.g., boldface)
```

```
[link text here](link.address.here)
```

```
![Alt text](/path/to/img.jpg "Optional title")
```

# Jekyll

1. Gather content from `_posts`, `_includes` and other files
2. Applies a template
3. Converts Markdown to HTML
4. Runs liquid converters
5. Outputs static HTML page

Note: Please note that Jekyll will only convert files that also have a YAML header at the top.



# YAML configuration

- ▶ Global configuration options: `__config.yml`  
Exposed from site
- ▶ Per page/post configuration options: `YAML Front Matter`  
Exposed from page



# Example



## A simple post

We place the blog entry inside `_posts` in a file called for example `_posts/2013-04-30-blog-like-hacker.md`

```
—  
layout: post  
title: Haskell  
title: Blogging Like a Hacker  
permalink: /blog/:year/:month/:day/:title  
menu:  
  - url:      http://jonan.org  
    description: my web page!  
  - url:      http://kestrelcluster.github.com  
    description: something amazing!
```

```
—  
  
# Hello world !
```

```
1. A list —:)
```

```
[A link to my web]({{ page.menu[0].url }})
```

```
![An image](/images/image.jpg)
```

<https://github.com/mojombo/jekyll/wiki/yaml-front-matter>

# The template

\_\_layout/post.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>{{ page.title }}</title>
  </head>
  <body>
    <ul>
      {% for item in page.menu %}
        <li><a href="{{item.url}}">{{item.title}}</a></li>
      {% endfor %}
    </ul>
    {{ content }}
  </body>
</html>
```



# Typical Workflow

- ▶ Commit to github :-)

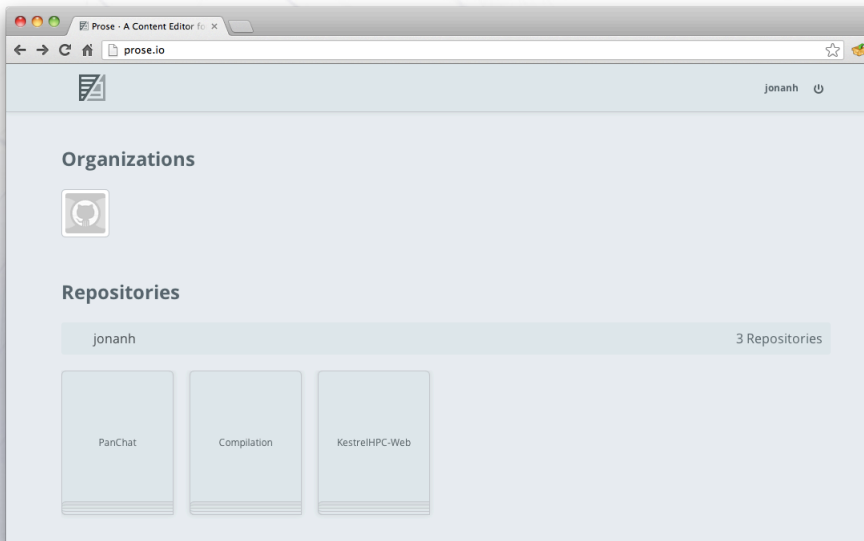
```
$ git add _posts/2013-04-30-blog-like-hacker.md
$ git commit -m "new post"
$ git push
```

- ▶ Run Jekyll locally.

```
jekyll --auto --server 8080
```

Each time a file is modified the whole site will be reconstructed.





The screenshot shows a web browser window with the address bar displaying "prose.io". The page title is "Prose - A Content Editor for...". The user's name "jonanh" is visible in the top right corner. The main content area is divided into two sections: "Organizations" and "Repositories". Under "Organizations", there is a GitHub logo. Under "Repositories", there is a header for the user "jonanh" with "3 Repositories" listed to the right. Below this header, three repository cards are displayed: "PanChat", "Compilation", and "KestrelHPC-Web".

Prose - A Content Editor for... x

← → ↻ 🏠 📄 prose.io ☆ 🍌

📄 jonanh ⏻

## Organizations

🐙

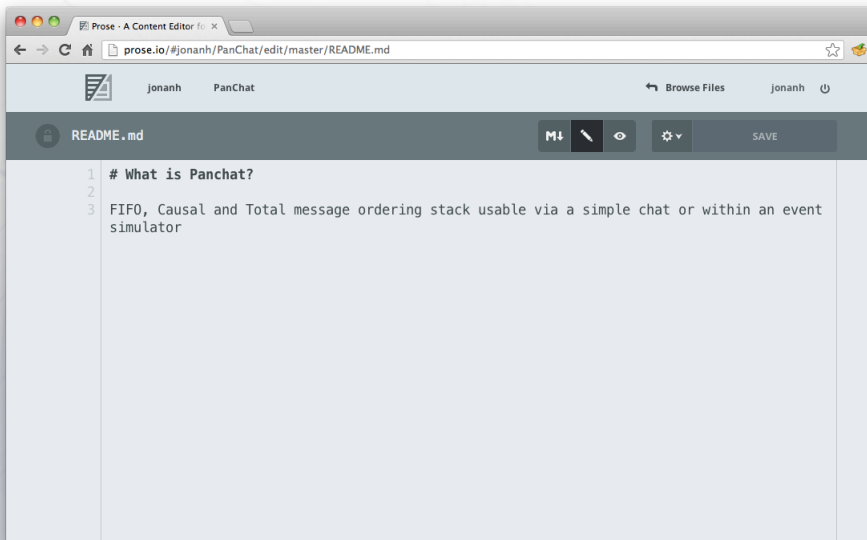
## Repositories

jonanh 3 Repositories

PanChat

Compilation

KestrelHPC-Web



The screenshot shows a web browser window with the address bar displaying `prose.io/#jonanh/PanChat/edit/master/README.md`. The browser's address bar also shows navigation icons (back, forward, refresh, home) and a star icon for bookmarks. Below the address bar, there is a navigation bar with a document icon, the text "jonanh PanChat", a "Browse Files" button, and a user profile icon for "jonanh".

The main editing area has a dark header bar with a lock icon, the filename "README.md", and several icons: a double arrow (undo/redo), a pencil (edit), an eye (visibility), a gear (settings), and a "SAVE" button. The text in the editor is as follows:

```
1 # What is Panchat?  
2  
3 FIFO, Causal and Total message ordering stack usable via a simple chat or within an event simulator
```

```
<!DOCTYPE HTML>
<html> <head>
<title>Blogging Like a Hacker</title> </head>
<body>
<ul>

  <li><a href="http://jonan.org"></a></li>

  <li><a href="http://kestrelcluster.github.com"></a></li>

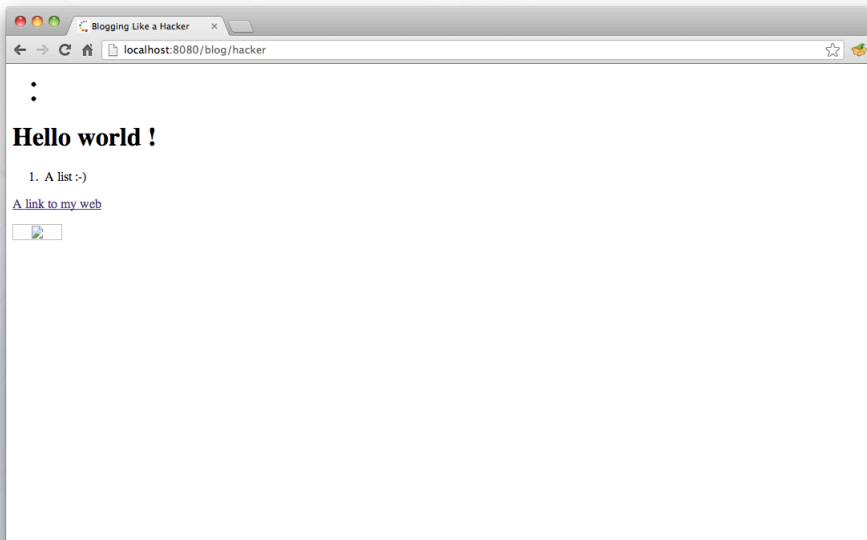
</ul>

<h1 id='hello_world_'>Hello world !</h1>

<ol>
<li>A list -:</li>
</ol>

<p><a href='http://jonan.org'>A link to my web</a></p>

<p><img alt='An image' src='/images/image.jpg' /></p>
</div>
</body>
</html>
```



## What if we want blog comments?

Extend with Disqus or Facebook comments



# Plugins

`_plugins/*.rb`



# Liquid Tag

This tags takes a parameter and looks for it on the context when rendered.

```
module Jekyll
  class ExampleTag < Liquid::Tag

    def initialize(tag_name, text, tokens)
      super
      @text = text.strip
    end

    def render(context)
      return context[@text]
    end
  end
end
Liquid::Template.register_tag('exampletag', Jekyll::ExampleTag)
```

```
{% exampletag site.globalvariable %}
```





# Liquid filter

```
module Jekyll
  module AssetFilter
    def asset_url(input)
      "http://www.example.com/#{input}?#{Time.now.to_i}"
    end
  end
end

Liquid::Template.register_filter(Jekyll::AssetFilter)
```



# Jekyll Conversor

```
module Jekyll
  module Converters
    class Markdown < Converter
      safe true

      def setup
        return KramdownParser.new @config
      end

      def matches(ext)
        ext =~ /md/i
      end

      def output_ext(ext)
        ".html"
      end

      def convert(content)
        parser = KramdownParser.new
        return parser.convert(content)
      end
    end
  end
end
```

# Jekyll Generator

```
module Jekyll

  class CategoryPage < Page
    def initialize(site, base, dir, category)
      @site, @base, @dir, @name = site, base, dir, 'index.html'

      self.process(@name)
      self.read_yaml(File.join(base, '_layouts'), 'category_index.html'←
      )
      self.data['category'] = category

      category_title_prefix = site.config['category_title_prefix'] || '←
      Category: '
      self.data['title'] = "#{category_title_prefix}#{category}"
    end
  end

  class CategoryPageGenerator < Generator
    safe true

    def generate(site)
      if site.layouts.key? 'category_index'
        dir = site.config['category_dir'] || 'categories'
        site.categories.keys.each do |category|
          site.pages << CategoryPage.new(site, site.source, File.join(←
          dir, category), category)
        end
      end
    end
  end
end
```

Questions?



Thanks!



# References

