

# Jekyll - Static Site Generator

ruby

Programming  
Language

ruby gem

Package Manager  
for ruby

⇒ Installing jekyll:

`$ gem install jekyll bundler`

★ Creating a site

`$ jekyll new <blog-name>`

⇒ Serving web site

`$ bundle exec jekyll serve`

→ bundle exec is needed only for the first time

→ From then on you can just type jekyll serve

★ Front Matter

→ Information about the page

→ All page will have a front matter on top.



## Example

---  
layout: post  
title: "New Post"  
date: 2017-08-20 15:58:59 -0700  
categories: jekyll update  
---

## \* Writing Posts

⇒ file naming convention:

`<date>-name.markdown`

or .md

## \* Working with Drafts

⇒ All drafts should be under `-drafts`

⇒ The draft posts will not show up on the Website.

⇒ To View draft posts run the server as

`$ jekyll serve --draft`

⇒ For draft you don't have to add date in front of it.

## \* Creating Pages

⇒ You will have two type of webpages:

① blog posts

② general pages



⇒ Example: About page

## \* Permalink

→ Permanent link for the page

→ In the front matter add

Permalink: "---/--"

⇒ You can construct permalink by accessing other variables.

### Example

permalink: /:category/:year/:month/:day/:title

## \* Event matter default

⇒ Inside the \_config.yml file add

defaults:

```
- Scope:
  path: ""
  type: "Posts"
  value:
    layout: "post"
    title: "My Title"
```

⇒ After changing \_config.yml you need to restart the jekyll server.



## ★ Theme

rubygems.org

↳ jekyll-theme

{ To search for  
new themes }

⇒ To install a theme:

↳ on Gemfile add

gem "<theme-name>"

↳ Then

\$bundle install

⇒ To add theme:

↳ on side -config.yml add

theme: "<theme-name>"

## ★ Layouts

↳ It is just a skeleton of HTML code that you can use to define look & feel for different kind of pages.

⇒ By default jekyll uses a theme.

↓  
{ theme has layout }

⇒ All layout lives -layouts folder.



**}} content }}**

→ In the layout.html file, this acts as the place holder of the content

⇒ There can be different levels of layout

→ A specific layout can have front matter specifying its layout.

→ {base layout}

## ★ Variables

**}} layout.<variable> }}**

→ Access variable in the layout.

**}} Page.<Variable> }}**

→ Access variable in the page that is using this layout.

**}} Site.<variable> }}**

→ Access variable from -config.xml file.



## \* Includes

↳ lives in `_includes`

⇒ Including a file:

```
{% include <file-name.html> %}
```

⇒ You can also pass information to includes:

```
{% include header.html color="blue" %}
```

You can access this in `header.html` file as  
`{{ include.color }}`

## \* Looping through posts

```
{% for post in site.posts %}
```

```
  {{ post.title }} <br>
```

```
{% endfor %}
```

## \* Conditionals

```
{% if page.title == "..." %}
```

```
  ...
```

```
{% elsif page.title == "..." %}
```

```
  ...
```

```
{% else %}
```

```
  ...
```

```
{% endif %}
```

You can combine conditionals  
with `and` & `or`



## ★ Data files (YAML, CSV)

↳ files live in \_data folder.

{% for person in site.data.people %}

{{ person.name }} <br>

{% endfor %}

## ★ Static Files

↳ Any file that don't have front matter.

## ★ Hosting on Github pages

1) edit \_config.yml file

↳ baseurl: <name of the repo>

2) Create a branch name gh-pages & add all the content to it.

3) Push the repo on github.

