# JEKYLL WORKSHOP

Promoted by NIAEFEUP

Created by Juliane Marubayashi

# WHY DO I NEED JEKYLL?

In case you you wanna have a cool blog or website to store your own content, Jekyll is a mantainable and easy tool to build it.

# INDEX

1. Install dependencies

2. Setup Jekyll

3. The first post

4. Diving into Jekyll basics

5. Setup theming

6. Deploy!

# 1. INSTALLING DEPENDENCIES

- Docker
- Docker Compose
- Github account

# 2. SETTING JEKYLL UP

Creating a Jekyll project as the website says require you to install a lot of stuff…

# So, it's better to use our friend...

## docker!

# 2.1 CREATE A PROJECT

Create a folder and execute inside the following command to setup the project.

```
docker run -v ${pwd}:/site bretfisher/jekyll new .  --force   # Windows
docker run -v $(pwd):/site bretfisher/jekyll new .  --force   # Linux
```

# 2.2 EXECUTE THE PROJECT

To start the project, execute the following command:

```
docker run -p 4000:4000 -v ${pwd}:/site bretfisher/jekyll-serve # Windows
docker run -p 4000:4000 -v $(pwd):/site bretfisher/jekyll-serve # Linux
```

Your awesome title

About

## Posts

May 24, 2022

Welcome to Jekyll!

subscribe via RSS

Your awesome title

Your awesome title

your-email@example.com

○ jekyll

♥ jekyllrb

Write an awesome description for your new site here. You can edit this line in _config.yml. It will appear in your document head meta (for Google search results) and in your feed.xml site description.

# 2.3 DOCKER COMPOSE

Writing the previous command everytime we want to
run our website can be painful. So, let's use docker
compose! Create a docker-compose.yml file.

```yaml
version: '2.4'

services:
  jekyll:
    image: bretfisher/jekyll-serve
    volumes:
      - .:/site
    ports:
      - '4000:4000'
```

# 2.4 RUNNING THE PROJECT WITH DOCKER COMPOSE

```
$ docker-compose up
```

Much easier, isn't it? :)

# Back to our agenda

1. Install dependencies

2. Setup Jekyll

3. **The first post**

4. Diving into Jekyll basics

5. Setup theming

6. Deploy!

# 3. THE FIRST POST

All the posts in Jekyll should be stored in the _posts folder in markdown format.

Only posts with the following syntax are considered:

yyyy-mm-dd-postname.md

# Each markdown should have a specific header to inherit the Jekyll style

```
1  ---
2  layout: post
3  title:  "This is my post!"
4  date:   2022-05-24 16:35:50 +0000
5  categories: jekyll first-post
6  ---
```

There aren't many options to put in the header, but you can check the other options in the website

# Back to our agenda

1. Install dependencies
2. Setup Jekyll
3. The first post
4. **Diving into Jekyll basics**
5. Setup theming
6. Deploy!

# 4. DIVING INTO JEKYLL BASICS

# 4.1 PAGES

We are able to create pages by adding new markdown files to the our root project.

If there're many files, we can organize it in subdirectories

For an website with a page to store subject notes we could have:

```
1  .
2  ├── subjects_list.md       # => http://example.com/subjects_list.
3  ├── programming_notes       # folder containing programming notes
4  │   ├── class1.md           # => http://example.com/programming_nc
5  │   └── class2.md           # => http://example.com/programming_nc
```

# 4.2 RESOURCES

Including resources is done as in conventional markdown

```
1  ![My helpful screenshot](/assets/screenshot.jpg)   // Display
2  [My pdf](/assets/mypdf.pdf)                // Adding a pdf link
```

reveal.js

# 4.3 LINK TO OTHER POSTS

Jekyll uses Liquid to reference variables and logical structures

In Liquid you output the content using double curly brackets {{variable}}

Logic statments are done with the {% %} syntax, e.g {% if statment %}

# Use the post_url syntax and reference post title without the file extension

```
1  [Random text]({% post_url 2022-05-24-my-post %})
```

# 4.4 REFERENCING OTHER PAGES

The {%link %} syntax acts like a joker. It can be used to retrieve posts, pages and also pdfs.

```
1  [link to a document]({% link _collection/name-of-document.md
2  [link to a post]({% link _posts/2016-07-26-name-of-post.md %
3  [link to a page]({% link news/index.html %})
4  [link to a file]({% link /assets/files/doc.pdf %})
```

If you are not sure of what is the path of your folder,
you can simply add {{page.path}} to any part of the
text, visualize it and then delete it again.

Yet, you can define some **variables** in jekyll by adding
the variable in the header and usign the {{}} syntax
to access it.

```
1  ---
2  title: My page
3  my_variable: footer_company_a.html
4  ---
5
6  {% link {{ page.my_variable }} %}
```

Everything that we have seem to far is available at [Jekyll's website](#)

If you are curious here is a list of the [variables available](#)

# Back to our agenda

1. Install dependencies

2. Setup Jekyll

3. The first post

4. Diving into Jekyll basics

5. **Setup theming**

6. Deploy!

# 5. SETTING UP A THEME

For this workshop we gonna setup the Minimal Mistakes Jekyll theme.

# 5.1 UPDATE GEM FILE

To install it update the Gemfile in your root folder, so
that the section jekyll_plugins and jekyll's version:

```
1  gem "jekyll"   # Before it was gem "jekyll", "~> 4.2.2"
2  group :jekyll_plugins do
3    gem "github-pages", group: :jekyll_plugins
4    gem "jekyll-include-cache", group: :jekyll_plugins
5  end
```

# 5.2 UPDATE_CONFIG.YML

**In your _config.yml you should:**

1. Add `jekyll-include-cache` in the plugins section
2. Add `remote_theme: "mmistakes/minimal-mistakes@4.24.0"`
3. Add the your github repository such as: repository: "NIAEFEUP/jekyll-workshop"
4. Remove any other `theme:` or `remote_theme:`

Your final file should be like this:

```yaml
1  title: Your awesome title
2  email: your-email@example.com
3  remote_theme: "mmistakes/minimal-mistakes@4.24.0"
4  description: >- # this means to ignore newlines until "base
5    Write an awesome description for your new site here. You
6    line in _config.yml. It will appear in your document head
7    Google search results) and in your feed.xml site descript
8  baseurl: "" # the subpath of your site, e.g. /blog
9  url: "" # the base hostname & protocol for your site, e.g.
10 twitter_username: jekyllrb
11 github_username: jekyll
12 repository: "NIAEFEUP/jekyll-workshop"
13
14 # Build settings
15 plugins:
```

# 5.3 RESTART DOCKER

Don't forget to stop docker after the alterations,
remove the images and build again!

```
1 docker image rm bretfisher/jekyll bretfisher/jekyll #
2 docker-compose up --build # Run and build docker
```

ur awesome title

## Recent Posts

### Welcome to Jekyll!

You'll find this post in your _posts directory. Go ahead and edit it and re-build the site to see your changes. You can rebuild the site in many different wa....

### My post!

A cool post.

**FEED**

022 Your awesome title. Powered by Jekyll & Minimal Mistakes.

# But now that we have changed the theme our posts are not rendering properly anymore!

Our layouts are not the same. Let's try to set our layout to single

```
1  ---
2  layout: single
3  title:  "My post!"
4  date:   2022-05-24 16:35:50 +0000
5  categories: jekyll update
6  ---
```

# Ufa! Everything sounds fine now! Except one thing...

# THE FONT!

As we zoom the screen the font increases and as we zoom out the font decreases

**This is not responsive!**

# 5.4 FIXING THE FONT

Create a file at assets/css/main.scss and copy the following code:

```scss
1   ---
2   # Only the main Sass file needs front matter (the dashes ar
3   ---
4
5   @charset "utf-8";
6
7   @import "minimal-mistakes/skins/{{ site.minimal_mistakes_sk
8   @import "minimal-mistakes";
9
10  html {
11      font-size: 16px; // change to whatever
12
13      @include breakpoint($medium) {
14          font-size: 18px; // change to whatever
15
```

# 5.5 RESOURCES

You can check the resources of the theme in the official website

# Back to our agenda

1. Install dependencies

2. Setup Jekyll

3. The first post

4. Diving into Jekyll basics

5. Setup theming

6. **Deploy!**

# 6. DEPLOY