Hugo

Hugo is a static site generator (*a software engine that uses text input files to generate static web pages*) written in Go.

- [Installation on MacOS](#)
- [Step-by-Step](#)

Installation on MacOS

This guide will walk you through setting up a development environment by installing Go, Dart Sass, and Hugo on your MacOS machine.

Prerequisites

Before installing Hugo, you'll need to have Go and Dart Sass installed on your system.

Installing Go

1. **Download the Go Package**:
   - Visit the Go Downloads page to download the latest version of Go suitable for your operating system.
2. **Install Go**:
   - Follow the installation instructions provided for your OS.
3. **Verify the Installation**:
   - Open a terminal and run the following command to check if Go is installed correctly:

   ```
   go version
   ```

   - You should see an output similar to:

   ```
   go version go1.22.4 darwin/amd64
   ```

Installing Dart Sass

1. **Install Dart Sass via Homebrew**:
   - Open a terminal and run the following command to install Dart Sass:

   ```
   brew install sass/sass/sass
   ```

2. **Verify the Installation**:
   - After the installation is complete, verify it by running:

   ```
   sass --version
   ```

   - The expected output should be:

   ```
   1.77.6
   ```

Installing Hugo

With Go and Dart Sass installed, you can now proceed to install Hugo.

1. **Install Hugo via Homebrew**:
   - In the terminal, run the following command:

   ```
   brew install hugo
   ```

2. **Verify the Installation**:
   - After installation, check that Hugo is installed correctly by running:

   ```
   hugo version
   ```

   - The output should be similar to:

   ```
   hugo v0.127.0+extended darwin/amd64 BuildDate=2024-06-05T10:27:59Z VendorInfo=brew
   ```

Conclusion

By following these steps, you've successfully set up Go, Dart Sass, and Hugo on your development environment. With these tools, you're now ready to start building and managing modern websites efficiently.

Happy me! 🎉

Step-by-Step

Hugo is a popular static site generator known for its speed and flexibility. This guide will help you get your site up and running in no time.

Step 1: Create Your Hugo Site Directory

To begin, you'll need to create the directory structure for your new Hugo site. Open your terminal and run the following command:

```
hugo new site site_directory
```

This command generates a new Hugo site in a directory named `site_directory`. You can replace `site_directory` with your preferred directory name.

Step 2: Navigate to the New Site Directory

Once the site structure is created, navigate to the newly created directory:

```
cd site_directory
```

This will be your working directory for the rest of the setup process.

Step 3: Choose and Install a Theme

Hugo relies on themes to style and structure your site. Browse the available themes on the Hugo Themes website and choose one that suits your needs.

After selecting a theme, download it into the `themes/` folder within your Hugo site directory. You can do this manually or by following the theme's installation instructions, typically found on its page.

Step 4: Configure Hugo to Use the Theme

To tell Hugo which theme to use, run the following command:

```
echo "theme = 'gallery'" >> hugo.toml
```

Replace `'gallery'` with the name of the theme you've chosen. This command appends the theme setting to your `hugo.toml` configuration file, making the theme active.

Step 5: Start the Development Server

With your theme in place, you can start the Hugo development server to see your site in action:

```
hugo server
```

Hugo will compile your site and start a local server. You can view your site in a browser at `http://localhost:1313`.

Step 6: Add a New Page

To add content to your site, create a new page by running:

```
hugo new content/about.md
```

This command creates a new Markdown file named `about.md` in the `content/` directory. You can edit this file to add text, images, and other content to your site.

Step 7: Edit the Site Configuration

Most themes come with specific configuration recommendations. Open your `hugo.toml` file and adjust the settings according to the theme author's guidelines. This might include setting up menus, configuring parameters, or adding social media links.

Step 8: Publish Your Site

Once you're satisfied with your site's content and configuration, you can build and publish your site. Run the following command:

```
hugo
```

This command compiles your site and generates static files in the `public/` directory. These files can be uploaded to any web server to make your site live.

Conclusion

By following these steps, you've successfully created a new Hugo site, chosen a theme, added content, and published your site. Hugo's simplicity and speed make it an excellent choice for building modern websites. Now that your site is live, you can continue to customize and expand it as needed.

Happy me! �ㅁ