In this tutorial, you'll learn how to use Gatsby themes by creating a new site using the official Gatsby blog theme.

Pre-requisites

- A Gatsby site

> *Note: This tutorial assumes you already have a Gatsby site to install your theme in. If you'd prefer to start with an entirely new site you can run* `gatsby new my-blog https://github.com/gatsbyjs/gatsby-starter-blog-theme` *to set up a starter with the blog theme already installed.*

Installing the blog theme

Navigate to the root of your project inside your terminal and install the theme with the following command:

```
npm install gatsby-theme-blog
```

Configure the theme

In your `gatsby-config.js` file, add `gatsby-theme-blog`. This theme takes optional dependencies that you can find in the README. However, you won't need to use them here.

```
module.exports = {
  plugins: [
    // highlight-start
    {
      resolve: "gatsby-theme-blog",
      options: {},
    },
    // highlight-end
    // you will likely have other plugins configured as well
  ],
}
```

> *Note: If you already have a landing page set up for your site, you may want to make use of the* `basePath` *option that will put your blog listing page at a path other than* `/`*, such as* `/blog`.

Update your site metadata

Customize the information on your site by replacing the site metadata in the `gatsby-config.js` file. Your `siteUrl` should point to your public domain. It's ok if you don't have one yet, you can update it later.

```
module.exports = {
  {/* highlight-start */}
  siteMetadata: {
    title: "My Blog",
    author: "Amberley Romo",
    description: "A collection of my thoughts and writings.",
    siteUrl: "https://amberley.blog/",
    social: [
      {
        name: "twitter",
        url: "https://twitter.com/amber1ey",
      },
      {
        name: "github",
        url: "https://github.com/amberleyromo",
      },
    ],
  },
  {/* highlight-end */}
  plugins: [
    {
      resolve: "gatsby-theme-blog",
      options: {},
    },
  ],
}
```

Add some content

Before you can see anything, you'll want to add some content so there is something to show.

By default, the posts are expected in the `/content/posts` directory, so create those folders and add a `my-post.md` file. Your file structure should look something like this.

```
my-blog
├── content
│   └── posts
│       └── my-post.md
├── src
├── gatsby-config.js
└── package.json
```

Despite the `md` extension, `my-post.md` is treated as an MDX file. When using this theme, you can use `md` and `mdx` extensions interchangeably.

Inside that Markdown file, add content. The top section is called [frontmatter]() and `title` and `date` are required fields.

~~~
---
title: My Post
date: 2020-04-15
---

Let's write a post!

```javascript
const test = "this is a theme"
```
~~~

Test run your site

To make sure everything is working, run your site. This command should be run in your terminal in your project's root directory.

```
gatsby develop
```

Navigate to `http://localhost:8000` to see the landing page of your site.

Replace your avatar

At the moment, the bio on your pages shows a blank section where a picture should be. Add your own avatar by choosing the image you want, and overwriting the file located at `/content/assets/avatar.png` .

Replace the content of the bio

When using Gatsby themes, you can take advantage of something called component shadowing. This allows you to override the default component included in the theme with a custom one you've created.

The Gatsby blog theme package has a component that contains the content of the site author's biography. The file path to that component (in the blog theme package, not your site) is `src/gatsby-theme-blog/components/bio-content.js` . You can find this path by looking through the theme in your site's `node_modules/gatsby-theme-blog` directory.

If you look at the file tree of your site, you'll see it looks something like this:

```
my-blog
├── content
│   ├── assets
│   │   └── avatar.png
│   └── posts
│       └── my-post.md
├── src
│   └── gatsby-theme-blog
│       └── components
│           └── bio-content.js
├── gatsby-config.js
└── package.json
```

In the `src` directory of the site, there's a `gatsby-theme-blog` directory. Any file placed in that directory with a path that matches the path of a file in the blog theme directory will completely shadow the theme.

> 💡 *The name of the directory (here `gatsby-theme-blog`) must exactly mirror the name of the published theme package, which in this case is `gatsby-theme-blog` .*

Open up the `bio-content.js` file and make some content edits:

```
import React, { Fragment } from "react"

export default function Bio() {
  return (
    {/* highlight-start */}
    <Fragment>
      This is my updated bio.
      <br />
      It's shadowing the content from the theme.
    </Fragment>
    {/* highlight-end */}
  )
}
```

At this point, you should have an updated avatar, updated site details, and an updated bio. You may want to re-run `gatsby develop` to make sure everything looks good.

Change the color theme

The blog theme uses `gatsby-plugin-theme-ui` to style your site. There are a number of presets available for you to make use of, or you can make your own!

> *If you want to use a preset take a look at the [Theme UI preset listing](#).*

You're going to use `@theme-ui/preset-funk` . To start, you have to install it.

```
npm install @theme-ui/preset-funk
```

Next, update your `gatsby-config.js` file to pass in the preset package name.

```
module.exports = {
  plugins: [
    {
      resolve: "gatsby-theme-blog",
      options: {
        preset: "@theme-ui/preset-funk", // highlight-line
      },
    },
  ],
}
```

If you want to further customize this theme you can shadow it. Create a file at `/src/gatsby-plugin-theme-ui/index.js` .

```
const darkBlue = `#007acc`
const lightBlue = `#66E0FF`
const blueGray = `#282c35`

export default {
  colors: {
    text: blueGray,
    primary: darkBlue,
    heading: blueGray,
  },
}
```

These colors will merge with the preset theme and override that part of the preset.

Change your prism theme

Another option you can make use of is prism styling for code blocks. There are many available from Theme UI.

In this example you'll use `prism-okaidia`. Update your `gatsby-config.js` file with that option.

```
module.exports = {
  plugins: [
    {
      resolve: "gatsby-theme-blog",
      options: {
        preset: "@theme-ui/preset-funk",
        prismPreset: "prism-okaidia", // highlight-line
      },
    },
  ],
}
```

When you restart your development server you'll see new syntax highlighting in your code snippets.

Take a look

Fire up your development server by running `gatsby develop` again in your terminal. Navigate to `http://localhost:8000` and take a look at your blog listing page.

Wrapping up

This was a step-by-step introduction to using a Gatsby theme through looking at a specific example. Note that different themes will be built differently, to accept different customization options. To dive deeper, check out the Gatsby Theme docs.

What's next?

- Using multiple themes together