

Gatsby Themes

Better, Faster, Flexible-er

What are Themes?

Themes?

- Token Swaps
- CSS Zen Garden

Wordpress/Hugo/etc

- Single Theme
- Comes with a visual style
- Adds Functionality

What are (Gatsby) Themes?

A new abstraction

(Gatsby) Themes?

- ~~Token Swaps~~
- CSS Zen Garden⁺⁺

WordPress/Hugo/etc ++

- ~~Single~~ Themes
- Comes with a visual style
- ~~Adds~~ composes Functionality

Why themes?

- Ease of use & higher level abstractions

Lifecycle & Action APIs

- onCreateNode
- createPage
- sourceNodes
- createTypes
- wrapRootElement
- etc...

Issues

- Plugins can't add plugins
- Low level abstractions

Who are themes for?

- People new to Gatsby
- People with common use cases
- Entrepreneurs
- Designers

Use Cases

1. I want a Blog
2. Documentation and a Blog
3. Customize the branding
4. Build Themes to sell
5. Internal Create React App
6. Data Sourcing

I want a Blog

```
yarn init -y  
yarn add gatsby-theme-blog
```

```
// gatsby-config.js  
module.exports = {  
  __experimentalThemes: ['gatsby-theme-blog']  
}
```

Docs and a Blog

```
yarn init -y
```

```
yarn add gatsby-theme-blog gatsby-theme-docz
```

Docs and a Blog

```
// gatsby-config.js
module.exports = {
  __experimentalThemes: [
    `gatsby-theme-blog`,
    `gatsby-theme-docz`
  ]
}
```



```
// gatsby-config.js
module.exports = {
  __experimentalThemes: [{
    resolve: `gatsby-theme-blog`,
    options: {
      colors: {
        black: '#000e1a',
        white: '#fff',
        blue: '#007ce0',
        navy: '#004175'
      }
    }
  }]
}
```

How do Themes work?

- gatsby-config.js
- Overrides and Configuration
- Data

```
// gatsby-config.js
module.exports = {
  plugins: [
    `gatsby-plugin-mdx`,
    `gatsby-plugin-emotion`,
    {
      resolve: `gatsby-source-filesystem`,
      options: {
        name: `pages`,
        path: `src/pages/`,
      },
    },
    `gatsby-source-airtable`,
  ]
}
```

```
// gatsby-config.js
module.exports = {
  __experimentalThemes: [`gatsby-theme-superawesome`],
  plugins: [
    `gatsby-source-airtable`,
  ]
}
```

```
// gatsby-theme-superawesome's gatsby-config.js
module.exports = {
  plugins: [
    `gatsby-plugin-mdx`,
    `gatsby-plugin-emotion`,
    {
      resolve: `gatsby-source-filesystem`,
      options: {
        name: `pages`,
        path: `src/pages/`,
      },
    },
  ],
}
```

Overrides and Configuration

- Configuring Themes
- Replacing components
- Extending components

Configurable Gatsby Config

```
// gatsby-config.js
module.exports = {
  __experimentalThemes: [{
    resolve: `gatsby-theme-superblog`,
    options: { tags: [`mdx`, `gatsby`, `pizza`] }
  }]
}
```

```
// A theme's gatsby-config.js
module.exports = options => {
  const optionalPlugins = []

  if(options.tags) {
    optionalPlugins.push({
      resolve: `gatsby-plugin-validate-blogpost`,
      options: { tags: options.tags }
    })
  }

  return {
    plugins
  }
}
```



```
// a theme's gatsby-node.js
exports.sourceNodes = ({actions}, themeOptions) => {
  actions.createTypes('enum Tag {'
    + themeOptions.tags.join('\n')
    + '}'
  )
}
```

```
// the generated code
enum Tag {
  mdx
  gatsby
  pizza
}
```

Expected a value of type "Tag" but received: "milk-duds"

Shadowing

```
.
└─ site
    ├── gatsby-config.js
    ├── package.json
    ├── node_modules
    │   └─ gatsby-theme-superawesome
    │       └─ src
    │           └─ components
    │               └─ blog-post-item.js
    └─ src
        └─ gatsby-theme-superawesome
            └─ components
                └─ blog-post-item.js
```

Shadowing

- Works for any webpack'd file in **src/**
- Direct replacement of file

```
// this file is our shadow file in our site
// site/src/gatsby-theme-superawesome/components/blog-post-item.js

// import original component
import BlogPostItem from 'gatsby-theme-superawesome/src/component

// don't show excerpts
export default props => (
  <BlogPostItem {...props} excerpt="" />
)
```

Latent Shadowing

1. Multiple themes
2. Shadowing
3. Shadowing themes that could be installed

When all else fails

- Low level APIs

Themes Roadmap and Other Projects

- Data Abstraction
- System UI
- DesignQL
- Theme UI
- Blocks

Data Abstraction

- Be able to use themes across sources
- Builds on Schema Customization APIs
- Is still early

System UI

- Design token format spec

- Schema customization and sharing data types across themes
- DesignQL, system-ui, CLI improvements, and other efforts of the themes team
- Advanced concepts such as latent shadowing, scoping styles and queries, and all-in-one or super composable themes