

---

---

# Ghostly Gatsby

— Intro to Gatsby.js  
with Alexx



# Overview

- Introduction to Gatsby
  - React
  - GraphQL
- Project Overview



# What is Gatsby?

- An open-source static site generator that has performance, scalability, and security built in
- Utilizes React for building the UI and GraphQL to provide a declarative development experience



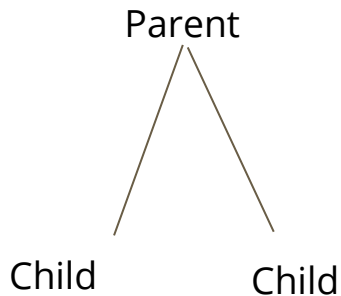
# Why did we choose Gatsby

- More than just a static site generator
  - Through the use of React and GraphQL
- The #1 fastest growing framework used by organizations of all shapes and sizes
- Can use popular frameworks and tools
- Easy to get started
- Open source -> Open Source -> Open Source -> Open Source



# React in 5 minutes

- A JavaScript framework used for building user interfaces
- Component-based
- Unidirectional (one-way) data flow -- data flows down from parent to child
- Types of data - state and props
- Lifecycle methods
- Virtual DOM - a diffing tool to check what to update
- Can be used for server-side rendering (SSR)



# Gatsby Runtime Datafetching via React

Class  
Component

```
import React from "react" 8.3K (gzipped: 3.3K)

class Example extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      data: []
    }
  }

  async componentDidMount() {
    const response = await fetch('url');
    this.setState({ data: response.json() });
  }

  render() {
    const { data } = this.state
    return (
      <div>
        {data.map((item, index) => (
          <ExampleItem
            key={index}
            title={item.title}
            description={item.short_description}
            url={item.link}
            topics={item.topics}
          />
        ))}
      </div>
    )
  }
}

const ExampleItem = ({ title, description, url, topics }) => (
  <div>
    <a target="_blank" rel="noopener noreferrer" href={url}>{title}</a>
    <p>{description}</p>
    <p>{topics.join(", ")}</p>
  </div>
)

export default Example
```

Function  
Component

```
import React, { useState, useEffect } from "react" 8.3K (gzipped: 3.3K)

const Example = () => {
  const [data, setData] = useState([])

  useEffect(() => {
    const fetchData = async () => {
      const response = await fetch('url');
      setData(response.data);
    };
    fetchData();
  }, []);

  return (
    <div>
      {data.map((item, index) => (
        <ExampleItem
          key={index}
          title={item.title}
          description={item.short_description}
          url={item.link}
          topics={item.topics}
        />
      ))}
    </div>
  )
}

const ExampleItem = ({ title, description, url, topics }) => (
  <div>
    <a target="_blank" rel="noopener noreferrer" href={url}>{title}</a>
    <p>{description}</p>
    <p>{topics.join(", ")}</p>
  </div>
)

export default Example
```



# GraphQL (briefly)

- A query language that helps you request exactly what you need
- Define the type and structure of your data
- Commonly used for APIs -- as opposed to REST
- In Gatsby -- used at buildtime; optional
  - Static queries
  - Page queries



Playground that ships with GraphQL for viewing queries





# Gatsby, React, GraphQL

```
module.exports = {
  siteMetadata: {
    title: 'Women Who Code',
    subTitle: 'Cincinnati',
    description: 'Women Who Code is a non-profit organization dedicated to inspiring women to excel in technology careers.',
    author: '@CincyWWCode',
    menuLinks: [
      {
        name: 'Members',
        link: '/members'
      },
      {
        name: 'Quotes',
        link: '/quotes'
      },
      {
        name: 'Resources',
        link: '/resources'
      },
      {
        name: 'Contact',
        link: '/contact'
      }
    ]
  }
}
```

```
import React from "react" 8.3K (gzipped: 3.3K)
import PropTypes from "prop-types" 1.7K (gzipped: 869)
import { useStaticQuery, graphql } from "gatsby"
import Header from "../header"
```

```
const Layout = ({ children }) => {
  const data = useStaticQuery(graphql`
    query SiteTitleQuery {
      site {
        siteMetadata {
          title
          subTitle
          menuLinks {
            name
            link
          }
        }
      }
    }
  `)

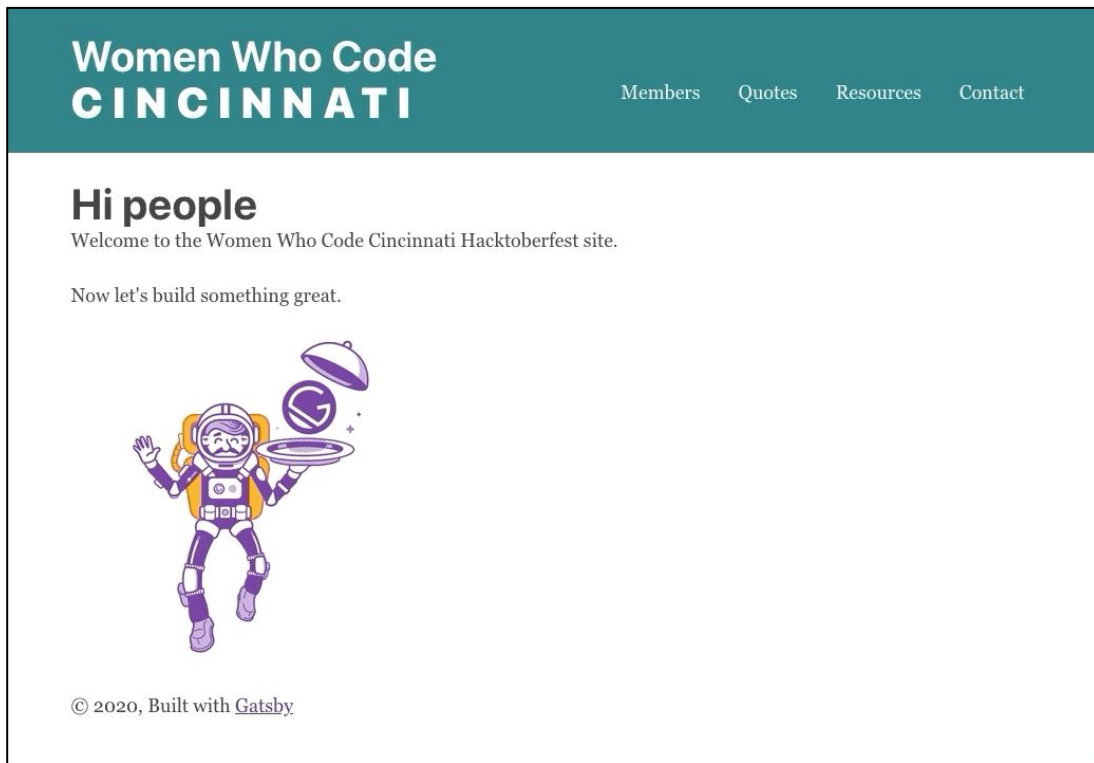
  return (
    <Header
      siteTitle={data.site.siteMetadata.title}
      subTitle={data.site.siteMetadata.subTitle}
      menuLinks={data.site.siteMetadata.menuLinks}
    />
    <div>
      <main>{children}</main>
      <footer>
        © {new Date().getFullYear()}, Built with
        {` `}
        <a href="https://www.gatsbyjs.com">Gatsby</a>
      </footer>
    </div>
  )
}

Layout.propTypes = {
  children: PropTypes.node.isRequired
}

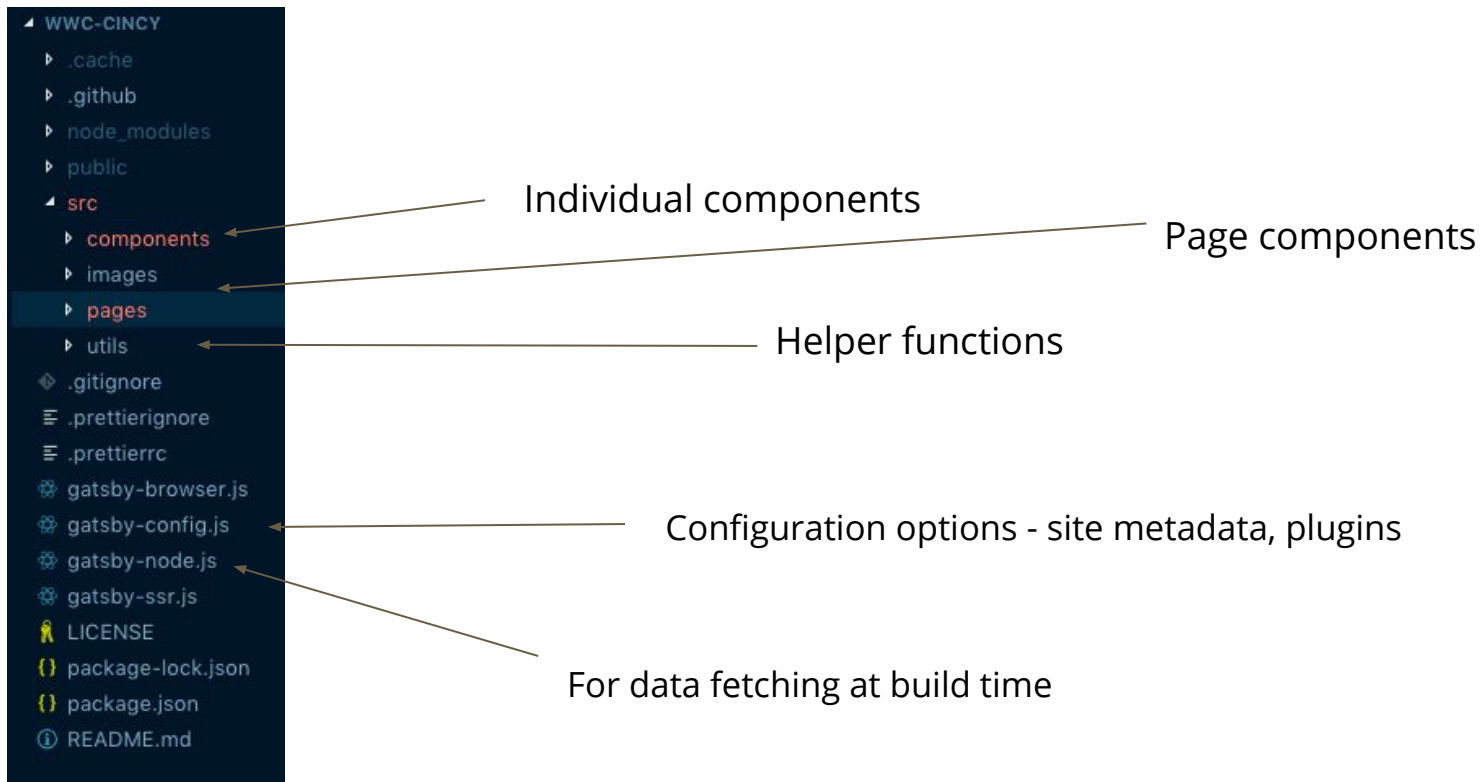
export default Layout
```



# What we're working on



# Project Structure



# Things to remember

Nothing is set in stone

- This repo is a working repository (things can change)

We are learning together and contributing to open source

