import HorizontalNavList from "@components/horizontal-nav-list"

When you're new to Gatsby there can be a lot of words to learn. This glossary aims to give you a 10,000-foot overview of common terms and what they mean for Gatsby sites.

<HorizontalNavList items={"ABCDEFGHIJKLMNOPQRSTUVWXYZ".split("")} slug={props.slug} />

A

AST

Abstract Syntax Tree: A tree representation of the source code that is found during a compilation step between two languages. For example, gatsby-transformer-remark will create an AST from Markdown to describe a Markdown document in a tree structure using the Remark parser.

API

Application Programming Interface: A method for one application to communicate with another. For example, a source plugin will often use an API to get its data.

Accessibility

The inclusive practice of removing barriers that prevent interaction with, or access to websites, by people with disabilities. When sites are correctly designed, developed and edited for accessibility, generally all users have equal access to information and functionality. Read about Gatsby's Commitment to Accessibility.

B

Babel

A tool that lets you write the most modern JavaScript, and during the build process it gets transpiled to code that most web browsers can understand.

Backend

The behind the scenes that the public does not see. This often refers to the control panel of your CMS. These are often powered by server-side programming languages such as Node.js, PHP, Go, ASP.net, Ruby, or Java.

Build

In Gatsby, this is the process of taking your code and content and packaging it into a website that can be hosted and accessed. Commonly referred to as *build time*. See also: backend and server-side.

C

Cache

A storage of information locally that might be used again, so computations and lookups can be retrieved faster from one place. Gatsby uses a cache to store information so it can build your site faster when you're developing without needing to do the same work twice.

CLI

Command Line Interface: An application that runs on your computer through the command line and interacted with your keyboard.

Gatsby has two command line interfaces. One, `gatsby`, for day-to-day development with Gatsby and another, `gatsby-dev`, for those who contribute to the Gatsby project.

Client-side

Client-side refers to operations that are performed by the user's browser in a client–server relationship in a computer network. In Gatsby, this is important when working with packages that rely on objects in the browser DOM, such as `window` or `navigator`. See also: server-side, frontend, and backend.

Client-side rendering

The practice of using JavaScript to render pages on the client-side, as opposed to server-side rendering alone. Gatsby uses React and the `@reach/router` library to enhance HTML pages compiled at build time to navigate site pages in a web browser without traditional page reloads, enabling performance techniques like preloading and pre-fetching, intersection observer and responsive `srcset` for images, and more. See also: routing, which is handled on the client-side in Gatsby by default.

CMS

Content Management System: an application where you can manage your content and have it saved to a database or file for accessing later. Examples of Content Management Systems include WordPress, Drupal, Contentful, and Netlify CMS.

Command Line

A text-based interface to run commands on your computer. The default Command Line applications for Mac and Windows are `Terminal` and `Command Prompt` respectively.

Compiler

A compiler is a program that translates code written in one language to another language. For example Gatsby can compile React applications into static HTML files. See also: transpile.

Component

Components are independent and re-usable chunks of code powered by React that, when combined, make up your website or app.

A component can include components within it. In fact, pages and templates are examples of components.

Config

The configuration file, `gatsby-config.js` / `gatsby-config.ts` tells Gatsby information about your website. A common option to set in this config is your site's metadata that can power your SEO meta tags.

Content Delivery Network

A content delivery network (CDN) is a highly distributed network of servers that stores copies of your content in locations that are closer to your site's visitors. Content delivery networks improve your site's performance by reducing the time needed to complete a network request.

Continuous Deployment

Continuous deployment (CD) automates the process of releasing changes to your project. A continuous deployment workflow automatically builds and tests your project, and publishes your changes only when they pass the required

tests.

CSS

CSS stands for Cascading Style Sheets, and it's a major part of the Web Platform with HTML and JavaScript. CSS is a language for styling webpages designed to be highly backwards-compatible. As new features are rolled out to end users, CSS parsers can safely ignore unsupported features and enhance with the properties they do support. CSS accomplishes this with its *cascading* design, fundamental to styling with new techniques like CSS Grid while providing fallbacks for older browsers. Gatsby supports multiple approaches to styling, including regular CSS files, CSS modules, and CSS-in-JS.

D

Data Source

Content and data's origin point, typically integrated into Gatsby with source plugins. A data source is often a Headless CMS, but it could also include Markdown, JSON, or YAML files.

Database

A database is a structured collection of data or content. Often a CMS will save to a database using backend technologies. They're often accessed in Gatsby via a source plugin

Decoupled

Decoupling describes the separation of different concerns. With Gatsby this most commonly means decoupling the frontend from the backend, like with Decoupled Drupal or Headless WordPress.

Decoupled Drupal

Decoupling refers to the practice of using Drupal as a headless CMS. A decoupled Drupal instance functions as a content API that returns JSON for your frontend to consume.

Deferred Static Generation (DSG)

Deferred Static Generation (DSG) is one of Gatsby's rendering options and allows you to defer non-critical page generation to user request, speeding up build times. Instead of generating every page at build time, you can decide to build certain pages up front and others only when a user accesses the page at run time.

Deploy

The process of building your website or app and uploading onto a hosting provider.

Development Environment

The environment when you're developing your code. It's accessed through the CLI using `gatsby develop`, and provides extra error reporting and things to help you debug before building for production.

DOM

The Document Object Model, commonly referred to as "the DOM", is a standard browser API that connects web pages to scripts or programming languages by representing the structure of an HTML document in memory. Developers commonly interact with the DOM through HTML markup (written in JSX in Gatsby), as well as both React and vanilla JavaScript code. Another important aspect of utilizing the DOM to its full potential is writing accessible HTML markup to expose a page's structure to assistive technology.

E

ECMAScript

ECMAScript (often referred to as ES) is a specification for scripting languages. JavaScript is an implementation of ECMAScript. Often developers will use Babel to transpile the latest ECMAScript code into more widely supported JavaScript.

Environment

The environment that Gatsby runs in. For example, when you are writing your code you probably want as much debugging as possible, but that's undesirable on the live website or app. As such, Gatsby can change its behavior depending on the environment it's in.

Gatsby supports two environments by default, the development environment and the production environment.

Environment Variables

Environment Variables allow you to customize the behavior of your app depending on its environment. For instance, you may wish to get content from a staging CMS during development and connect to your production CMS when you build your site. With environment variables you can set a different URL for each environment.

F

Filesystem

The way files are organized. With Gatsby, it means having files in the same place as your website's or app's code instead of pulling data from an external source. Common filesystem usage in Gatsby includes Markdown content, images, data files, and other assets.

Frontend

The public-facing interface for your website or app, delivered using web technologies: HTML, CSS, and JavaScript. For more insight into how the Web Platform brings these technologies together, check out this article on How Browsers Work.

G

Gatsby

Gatsby is a modern website framework that builds performance into every website or app by leveraging the latest web technologies such as React, GraphQL, and modern JavaScript. Gatsby makes it easy to create blazing fast, compelling web experiences without needing to become a performance expert.

GraphQL

A query language that allows you to pull data into your website or app. It's the interface that Gatsby uses for managing site data.

H

HTML

A markup language that every web browser is able to understand. It stands for Hypertext Markup Language. HTML gives your web content a universal informational structure, defining things like headings, paragraphs, and more. It is also key to providing an accessible website.

Headless CMS

A CMS that only handles the backend content management instead of handling both the backend and frontend. This type of setup is also referred to as Decoupled.

Headless WordPress

The practice of using JSON returned from the WordPress REST API as a headless CMS. It allows you to use WordPress to write and edit content that can be consumed by any client capable of parsing JSON.

Hosting

A hosting provider keeps a copy of your website or app and makes it accessible to the public. Common hosting providers for Gatsby projects include Netlify, AWS, S3, Surge, Heroku, and more.

Hot module replacement

A feature in use when you run `gatsby develop` that live updates your site on save of code in a text editor by automatically replacing modules, or chunks of code, in an open browser window. Gatsby uses Fast Refresh.

Hydration

Once a site has been built by Gatsby and loaded in a web browser, client-side JavaScript assets will download and turn the site into a full React application that can manipulate the DOM. This process is often called re-hydration as it runs some of the same JavaScript code used to generate Gatsby pages, but this time with browser DOM APIs like `window` available.

I

Inference

As part of its data layer and build process, Gatsby will automatically **infer** a schema, or type-based structure, based on available data sources (e.g. Markdown file nodes, WordPress posts, etc.). More control can be gained over this structure by using Gatsby's Schema Customization API.

Infrastructure as Code

Infrastructure as Code is the practice of using configuration files and scripts to automate the process of setting up your development, testing, and production environments.

J

JAMStack

JAMStack refers to a modern web architecture using JavaScript, APIs, and (HTML) markup. From JAMStack.org: "It's a new way of building websites and apps that delivers better performance, higher security, lower cost of scaling, and a better developer experience."

JavaScript

A programming language that helps us make the web dynamic and interactive. JavaScript is a widely deployed web technology in browsers. It is also used on the server-side with Node.js. It is an implementation of the ECMAScript specification.

JSX

JSX is an extension to JavaScript that allows developers to write HTML and custom components in the same piece of code. The React team recommends using it to describe what a UI should look like. JSX may remind you of a template language, but it comes with the full power of JavaScript. Some important details to note are that because JSX uses JavaScript, some HTML attributes in your markup have to be swapped out to avoid reserved words in JavaScript (things like `htmlFor` and `className`).

K

L

Linting

Linting is the process of running a program that will analyze code for potential errors. The Gatsby project uses prettier to identify and fix common style issues. Another example of a linter commonly used in React projects is eslint-plugin-jsx-a11y, which checks for common accessibility issues in development.

M

MDX

Extends Markdown to support React components within your content.

Markdown

A way of writing HTML content with plain text, using special characters to denote content types such as hash symbols for headings, and underscores and asterisks for text emphasis.

N

npm

Node package manager. Allows you to install and update other packages that your project depends on. Gatsby and React are examples of your project's dependencies. See also: Yarn.

Node

Gatsby uses data nodes to represent a single piece of data. A data source will create multiple nodes.

Node.js

A program that lets you run JavaScript on your computer. Gatsby is powered by Node.

O

P

Package

A package usually describes a JavaScript program that has additional information about how it should be distributed and used, such as its version number. npm and Yarn manages and installs the packages your project uses. Gatsby itself is a package.

Page

An HTML page.

This also often refers to components that live in `/src/pages/` and are converted to pages by Gatsby, as well as pages created dynamically in your `gatsby-node.js` file.

Plugin

Additional code that adds functionality to Gatsby that wasn't included out-of-the-box. Common Gatsby plugins include source and transformer plugins for pulling in and manipulating data, respectively.

Production Environment

The environment for the built website or app that users will experience when deployed. It can be accessed through the CLI using `gatsby build` or `gatsby serve`.

Programmatically

Something that automatically happens based on your code and configuration. For example, you might configure your project to create a page for every blog post written, or read and display the current year as part of a copyright in your site footer.

Progressive Enhancement

Progressive enhancement is a strategy for the web that emphasizes core page content is loaded from a server before anything else, without JavaScript as a requirement to load. This strategy then progressively adds more complex layers of presentation and features on top of the content as the end user's browser/network connection allow. Gatsby's default approach to building pages ahead-of-time means content will load first and enhance as scripts download and execute.

Public

This usually refers to either a member of the public (as opposed to your team) or the folder `/public` in which your built website or app is saved.

Q

Query

The process of requesting specific data from somewhere. With Gatsby you normally query with GraphQL.

R

React

A code library (written with JavaScript) for building user interfaces. It's the framework that Gatsby uses to build pages and structure content.

Remark

A parser to translate Markdown to other formats like HTML or React code.

Runtime

Runtime is when a program is running (or being executable); it can refer to a few things. Node.js is a server-side runtime that executes JavaScript code. Client-side JavaScript, on the other hand, refers to the browser runtime where traditional JavaScript code executes. Gatsby compiles your site at build time and rehydrates with a React runtime to provide a fast, interactive, and dynamic user experience.

Routing

Routing is the mechanism for loading the correct content in a website or app based on a network request - usually a URL. For example, it allows for routing URLs like `/about-us` to the appropriate page, template, or component.

S

Schema

An exact representation of how data is stored in a system, such as tables and fields in a database or a JSON file structure. In Gatsby, the GraphQL schema expresses all queryable data - or data that components can ask about as part of Gatsby's data layer.

Server-side

The server-side part of the client-server relationship refers to operations performed by a computer program which manages access to a centralized resource or service in a computer network. See also: frontend and backend.

Server-side rendering

Using a Node.js-based server to generate HTML in response to a request from a user agent such as a browser. Gatsby uses the server-side technology Node.js to compile pages at build time, as opposed to serving them at browser runtime with client-side JavaScript.

Source Code

Source code is your code that lives in `/src/` folder and makes up the unique aspects of your website or app. It is made up of JavaScript and sometimes CSS and other files.

The source code gets built into the site the public will see.

Source Plugin

A plugin that adds additional data sources to Gatsby that can then be queried by your pages and components.

Starter

A pre-configured Gatsby project that can be used as a starting point for your project. They can be discovered using the Gatsby Starter Library and installed using the Gatsby CLI.

Static

Gatsby builds static versions of your page that can be easily hosted. This is in contrast to dynamic systems in which each page is generated on-the-fly. Being static affords major performance gains because the work only needs to be

done once per content or code change.

It also refers to the `/static` folder which is automatically copied into `/public` on each build for files that don't need to be processed by Gatsby but do need to exist in public.

Static Site Generator

A software application that creates HTML pages from templates or components and a given content source.

T

Template

A component that is programmatically turned into a page by Gatsby.

Theme

A Gatsby theme is like a WordPress theme that is composable (with other themes), extendable (with more logic), and replaceable (shadowing). Gatsby themes can have any facet of a Gatsby app packaged inside of them, and can also offer any number of knobs to turn features on or off.

Transformer

A plugin that transforms one type of data to another. For example you might transform a spreadsheet into a JavaScript array.

Transpile

The process of converting code from one syntax or format to another, such as TypeScript, a superset of JavaScript which provides custom type checking during development. Babel is another common example of transpilation that reformats newer JavaScript code following the ECMAScript standard to be more backwards-compatible during the site compilation process.

U

UI

A UI refers to a User Interface. In the field of human-computer interaction, a UI is a space where interactions between humans and machines occur. The goal of this interaction is to allow effective operation and control of the machine from the human end, while the machine simultaneously feeds back information that aids the user's decision-making process (such as error messages or notifications).

V

W

webpack

A JavaScript application that Gatsby uses to bundle your website's code up. This happens automatically on build.

WPGraphQL

A WordPress plugin that adds GraphQL capabilities to WordPress. It's another way that you can use WordPress as a content source for Gatsby.

X

Y

Yarn

A package manager that some prefer to npm. It is also required for developing Gatsby.

Z