

凡例	評価	評価内容
★	優	完全に利用可能
○	良	プラグインなどで一部利用可能
△	可	カスタマイズが必要、または限定的
×	不可	実現不可能

Category01	Category02	Category03	Category Description	Next.js	Hugo	Gatsby	Jekyll	Nuxt.js
Performance	Delivery Optimization	Static content	Build your site as static files which can be deployed without a server.	★	★	★	★	★
		CDN	Cache your site on a CDN distributed throughout the globe.	○	○	○	○	○
		AMP support	Use Google's AMP's static page format, which enables faster loads and are privileged in Google search results.	○	○	○	○	○
	Payload optimization	Progressive image loading	Load images progressively by displaying a blurry placeholder image before loading the full heavyweight asset, thus preventing pages from appearing incomplete.	○	△	★	△	○
		Responsive image loading	Enable modern browsers to load the right size of image assets given viewport size through responsive images.	○	○	★	○	○
		Inlines critical CSS	Inline CSS within your rendered HTML in order to prevent blocking page load as occurs in the case of a separate CSS file.	★	△	★	△	○
		Font self-hosting	Host fonts on your own server so you can load them from your local filesystem so that a page load-blocking request to external servers does not occur.	★	★	★	★	★
	Progressive Web App	Offline access	Provide offline access via service workers in Progressive Web Apps so that users on spotty connections can access your site.	★	○	★	○	○
		Prefetching of linked pages	Prefetch all the content needed to load the next link you click in the background while you browse the page as it loads.	★	△	★	△	★
		Page caching	Allow browsers to serve content locally without an extra network call when a user visits a previously visited page by fingerprinting static resources that aren't expected to change.	★	★	★	★	★
		No extraneous code fetching	Fetch only the code for the loaded page on page load as opposed to the code needed to run the entire application.	★	★	★	★	★
Developer Experience	Maintainability and debuggability	Serverless features	Reduce operations overhead by using serverless hosting that doesn't require security or framework upgrades or accessing a live production server.	★	★	★	★	★
		Preview on refresh or with link	Click a Preview button or refresh the page to preview what your content looks like while writing or editing a piece of content.	★	★	★	★	★
		Hot reloading of content	See UI changes without refreshing the page in your local development environment browser while editing content.	★	★	★	★	★
		Hot reloading of code	See UI changes without refreshing the page in your local development environment browser while editing code.	★	★	★	★	★
	Declarative rendering	Componentization	Plug and play internal or external third-party components from a shared codebase or component library.	★	△	★	△	★
		One-way data binding	Use a unidirectional data flow without complex interdependencies that are present in other data flow approaches like MVC in order to build complex front-end components.	★	★	★	★	★
		Declarative data queries	Use declarative queries in GraphQL that are colocated with your components to retrieve only certain data in a certain shape. Gatsby also pulls data into the GraphQL API so you can query it in your local IDE or query constructor.	★	×	★	×	★
		Declarative UI building	Use a hierarchical UI approach by declaratively passing properties down child trees in React, with faster debug cycles through the DevTools extension and element-by-element state comparisons.	★	×	★	×	★
	Modern development practices	Asset pipelines	Use asset pipelines, the compilers of the web stack, to transform images and discrete code into minified, raw HTML, CSS, and JavaScript that browsers can parse.	★	★	★	△	★
		CSS Extensions (e.g. Sass)	Use languages like Sass and LESS and libraries like Glamor and Aphrodite to colocate CSS with JavaScript and HTML in React in order to solve common problems in vanilla CSS like global namespacing, nondeterminism, dead code elimination, and minification.	★	★	★	★	★
		Modern JavaScript syntax	Access modern JavaScript features including syntactic sugar, new language features, new data structures and types, and modularity.	★	×	★	△	★

凡例	評価	評価内容
★	優	完全に利用可能
○	良	プラグインなどで一部利用可能
△	可	カスタマイズが必要、または限定的
×	不可	実現不可能

Category01	Category02	Category03	Category Description	Next.js	Hugo	Gatsby	Jekyll	Nuxt.js
Governance	Security	XSS protection	Protect your application from cross-site scripting (XSS) vulnerabilities that could expose your site to attackers.	★	★	★	★	★
		Access control (CORS)	Enable access control through cross-origin resource sharing (CORS) to prevent unwanted access through vectors such as distributed-denial-of-service (DDoS) attacks.	★	★	★	★	★
		Environment variables	Use environment variables so you can deploy configuration to hosting services without exposing them in your codebase.	★	★	★	★	★
		User authentication	Provide user authentication through an authentication provider that will permit access to sensitive areas of your application.	○	○	○	○	○
	Design	Programmatic design	Use tools like Typography.js to make it easier to programmatically design your site by generating CSS that adheres to a certain intended design.	○	×	○	×	×
		Design systems	Export application components for use in other applications such as Sketch, which has integrations like react-sketchapp for React components.	○	×	○	×	○
		Component libraries	Leverage component libraries in other applications like Sketch with integrations like react-sketchapp to accelerate the articulation of your style guide.	○	×	○	×	○
Accessibility	Accessible markup	Progressive enhancement	Provide a baseline of progressive enhancement where all sites can be accessed as static HTML and CSS and where all fundamental functionality is preserved without JavaScript being enabled.	★	★	★	★	★
		Landmark elements	Leverage ARIA landmarks to provide semantic markers that indicate certain page elements for users of assistive and accessible technologies such as screen readers.	○	○	○	○	○
		Semantic HTML	Introduce meaning to HTML to reinforce desired semantics and to aid the navigation of your site by users of assistive and accessible technologies such as screen readers.	○	○	○	○	○
	Accessible defaults	Accessible routing	Provide focus management on each page change through libraries such as @reach/router, an accessible routing library.	△	○	★	○	×
		Accessibility statement	Leverage a framework with a clear and robust accessibility statement explaining its position on accessibility and that it is a first-class consideration.	×	×	★	×	×
		WCAG 2.0 AA compliance	Leverage a framework with full compliance with the Web Content Accessibility Guidelines (WCAG) 2.0. Gatsby is partially compliant.	△	○	○	○	△

凡例	評価	評価内容
★	優	完全に利用可能
○	良	プラグインなどで一部利用可能
△	可	カスタマイズが必要、または限定的
×	不可	実現不可能

Category01	Category02	Category03	Category Description	Next.js	Hugo	Gatsby	Jekyll	Nuxt.js
Documentation	Tutorials and guides	Introductory tutorial	Read an introductory tutorial with a step-by-step process to help new users get started quickly.	★	★	★	★	★
		Sourcing data	Read a guide with a step-by-step process to help you source data from a data source such as a CMS.	△	△	★	△	△
		Incorporating CSS libraries	Read a guide with a set of instructions to incorporate a CSS preprocessor such as Sass, LESS, or Stylus; and CSS-in-JS libraries like Glamor, or Aphrodite.	★	★	★	★	★
	Development guides	Routing	Read a guide with a set of instructions to perform routing and route management in the framework.	★	★	★	★	★
		Testing	Read a guide with a set of instructions to perform testing and quality assurance such as linting within a development workflow.	△	△	★	★	★
		Debugging	Read a guide with a set of instructions to perform debugging and troubleshooting within a development workflow.	△	★	★	○	△
		Performance	Read a guide with a set of instructions to improve performance benchmarks within the framework.	△	★	★	×	○
	Feature addition guides	Adding media assets	Read a guide with a set of instructions to add media assets such as image and video to your application.	○	★	★	★	★
		Adding search	Read a guide with a set of instructions to add search capabilities and indexing to your application.	△	★	★	△	△
		Adding analytics	Read a guide with a set of instructions to add analytics integration and segmentation to your application.	△	★	★	○	★
		Adding authentication	Read a guide with a set of instructions to add authentication providers to gate access to your application.	△	○	★	○	★
		Adding SEO	Read a guide with a set of instructions to perform search engine optimization for your application.	△	△	★	○	○
Ecosystem	Ecosystem	Component ecosystem	Leverage an existing component ecosystem such as React's, which includes out-of-the-box component libraries and curated sets such as JSCoach.	★	×	★	×	★
		Hosted option	Plug your application into static hosts such as Netlify, Render, or surge.sh. WordPress and Squarespace include built-in hosting.	○	○	○	○	○
		Themes ecosystem	Leverage a theme ecosystem with various options for your application to take on a custom look and feel. Jekyll has themes, and WordPress and Squarespace offer default theme selection.	△	★	○	★	△
	Integrations	OOTB integrations with 20+ other systems	Leverage a framework that has integrations off the shelf with more than 20 other systems.	×	×	★	×	×
		OOTB integrations with 50+ other systems	Leverage a framework that has integrations off the shelf with more than 50 other systems.	×	×	★	×	×
	Community	Swag store	Purchase items from a swag store to express your support of the framework community.	×	×	★	×	★
		Code of conduct	Feel welcome and empowered to contribute within a community that has an official code of conduct governing all interactions.	★	×	★	★	★
		Contributing section in documentation	Understand how to contribute to a community with a clear section in documentation for new contributors.	★	★	★	★	★