

Apache Doris 1.1 (English)



Table of contents:

- [Tutorial Intro](#)
 - [Getting Started](#)
 - [Generate a new site](#)
 - [Start your site](#)
- [Create a Page](#)
 - [Create your first React Page](#)
 - [Create your first Markdown Page](#)
- [Create a Document](#)
 - [Create your first Doc](#)
 - [Configure the Sidebar](#)
- [Create a Blog Post](#)
 - [Create your first Post](#)
- [Markdown Features](#)
 - [Front Matter](#)
 - [Links](#)
 - [Images](#)
 - [Code Blocks](#)
 - [Admonitions](#)
 - [MDX and React Components](#)
- [Deploy your site](#)
 - [Build your site](#)
 - [Deploy your site](#)
- [Congratulations!](#)
 - [What's next?](#)
- [Manage Docs Versions](#)
 - [Create a docs version](#)
 - [Add a Version Dropdown](#)
 - [Update an existing version](#)
- [Translate your site](#)
 - [Configure i18n](#)
 - [Translate a doc](#)
 - [Start your localized site](#)
 - [Add a Locale Dropdown](#)
 - [Build your localized site](#)



Tutorial Intro

Let's discover **Docusaurus** in less than 5 minutes.

Getting Started

Get started by **creating a new site**.

Or **try Docusaurus immediately** with new.docusaurus.io.

Generate a new site

Generate a new Docusaurus site using the **classic template**:

```
npx @docusaurus/init@latest init my-website classic
```

Start your site

Run the development server:

```
cd my-website
```

```
npx docusaurus start
```

Your site starts at `http://localhost:3000`.

Open `docs/intro.md` and edit some lines: the site **reloads automatically** and display your changes.



Create a Page

Add **Markdown or React** files to `src/pages` to create a **standalone page**:

- `src/pages/index.js` -> `localhost:3000/`
- `src/pages/foo.md` -> `localhost:3000/foo`
- `src/pages/foo/bar.js` -> `localhost:3000/foo/bar`

Create your first React Page

Create a file at `src/pages/my-react-page.js`:

`src/pages/my-react-page.js`

```
import React from 'react';
import Layout from '@theme/Layout';

export default function MyReactPage() {
  return (
    <Layout>
      <h1>My React page</h1>
      <p>This is a React page</p>
    </Layout>
  );
}
```

A new page is now available at `http://localhost:3000/my-react-page`.

Create your first Markdown Page

Create a file at `src/pages/my-markdown-page.md`:

`src/pages/my-markdown-page.md`

```
# My Markdown page
```

This is a Markdown page

A new page is now available at <http://localhost:3000/my-markdown-page>.



Create a Document

Documents are **groups of pages** connected through:

- a **sidebar**
- **previous/next navigation**
- **versioning**

Create your first Doc

Create a markdown file at `docs/hello.md`:

```
docs/hello.md
```

```
# Hello
```

```
This is my **first Docusaurus document**!
```

A new document is now available at `http://localhost:3000/docs/hello`.

Configure the Sidebar

Docusaurus automatically **creates a sidebar** from the `docs` folder.

Add metadatas to customize the sidebar label and position:

```
docs/hello.md
```

```
---  
sidebar_label: 'Hi!'  
sidebar_position: 3  
---
```

```
# Hello
```

```
This is my **first Docusaurus document**!
```

It is also possible to create your sidebar explicitly in `sidebars.js`:

sidebars.js

```
module.exports = {
  tutorialSidebar: [
    {
      type: 'category',
      label: 'Tutorial',
      - items: [...],
      + items: ['hello'],
    },
  ],
};
```



Create a Blog Post

Docusaurus creates a **page for each blog post**, but also a **blog index page**, a **tag system**, an **RSS** feed...

Create your first Post

Create a file at `blog/2021-02-28-greetings.md`:

```
blog/2021-02-28-greetings.md
```

```
---
slug: greetings
title: Greetings!
author: Steven Hansel
author_title: Docusaurus Contributor
author_url: https://github.com/ShinteiMai
author_image_url: https://github.com/ShinteiMai.png
tags: [greetings]
---
```

Congratulations, you have made your first post!

Feel free to play around and edit this post as much you like.

A new blog post is now available at `http://localhost:3000/blog/greetings`.



Markdown Features

Docusaurus supports **Markdown** and a few **additional features**.

Front Matter

Markdown documents have metadata at the top called **Front Matter**:

my-doc.md

```
---
id: my-doc-id
title: My document title
description: My document description
slug: /my-custom-url
---

## Markdown heading

Markdown text with [links](./hello.md)
```

Links

Regular Markdown links are supported, using url paths or relative file paths.

Let's see how to [\[Create a page\]\(/create-a-page\)](#).

Let's see how to [\[Create a page\]\(./create-a-page.md\)](#).

Result: Let's see how to [Create a page](#).

Images

Regular Markdown images are supported.

Add an image at `static/img/docusaurus.png` and display it in Markdown:

```
![Docusaurus logo](/img/docusaurus.png)
```



Code Blocks

Markdown code blocks are supported with Syntax highlighting.

```
```jsx title="src/components/HelloDocusaurus.js"
function HelloDocusaurus() {
 return (
 <h1>Hello, Docusaurus!</h1>
)
}
```
```

src/components/HelloDocusaurus.js

```
function HelloDocusaurus() {
  return <h1>Hello, Docusaurus!</h1>;
}
```

Admonitions

Docusaurus has a special syntax to create admonitions and callouts:

:::tip My tip

Use this awesome feature option

:::

:::danger Take care

This action is dangerous

:::

MY TIP

Use this awesome feature option

TAKE CARE

This action is dangerous

MDX and React Components

MDX can make your documentation more **interactive** and allows using any **React components inside Markdown**:

```
export const Highlight = ({children, color}) => (  
  <span  
    style={{  
      backgroundColor: color,  
      borderRadius: '20px',  
      color: '#fff',  
      padding: '10px',  
      cursor: 'pointer',  
    }}  
    onClick={() => {  
      alert(`You clicked the color ${color} with label ${children}`)  
    }}  
  >  
    {children}  
  </span>  
)  
);
```

This is <Highlight color="#25c2a0">Docusaurus green</Highlight> !

This is <Highlight color="#1877F2">Facebook blue</Highlight> !

This is Docusaurus green !

This is Facebook blue !



Deploy your site

Docusaurus is a **static-site-generator** (also called **Jamstack**).

It builds your site as simple **static HTML, JavaScript and CSS files**.

Build your site

Build your site **for production**:

```
npm run build
```

The static files are generated in the `build` folder.

Deploy your site

Test your production build locally:

```
npm run serve
```

The `build` folder is now served at `http://localhost:3000/`.

You can now deploy the `build` folder **almost anywhere** easily, **for free** or very small cost (read the **Deployment Guide**).



Congratulations!

You have just learned the **basics of Docusaurus** and made some changes to the **initial template**.

Docusaurus has **much more to offer!**

Have **5 more minutes**? Take a look at [versioning](#) and [i18n](#).

Anything **unclear** or **buggy** in this tutorial? [Please report it!](#)

What's next?

- Read the [official documentation](#).
- Add a custom [Design and Layout](#)
- Add a [search bar](#)
- Find inspirations in the [Docusaurus showcase](#)
- Get involved in the [Docusaurus Community](#)



Manage Docs Versions

Docusaurus can manage multiple versions of your docs.

Create a docs version

Release a version 1.0 of your project:

```
npm run docusaurus docs:version 1.0
```

The `docs` folder is copied into `versioned_docs/version-1.0` and `versions.json` is created.

Your docs now have 2 versions:

- `1.0` at `http://localhost:3000/docs/` for the version 1.0 docs
- `current` at `http://localhost:3000/docs/next/` for the **upcoming, unreleased docs**

Add a Version Dropdown

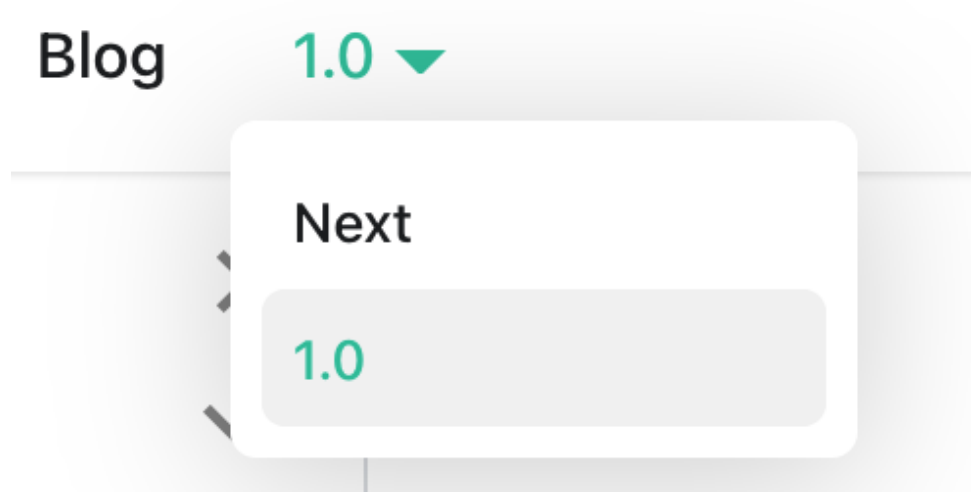
To navigate seamlessly across versions, add a version dropdown.

Modify the `docusaurus.config.js` file:

`docusaurus.config.js`

```
module.exports = {
  themeConfig: {
    navbar: {
      items: [
        {
          type: 'docsVersionDropdown',
        },
      ],
    },
  },
};
```

The docs version dropdown appears in your navbar:



Update an existing version

It is possible to edit versioned docs in their respective folder:

- `versioned_docs/version-1.0/hello.md` updates `http://localhost:3000/docs/hello`
- `docs/hello.md` updates `http://localhost:3000/docs/next/hello`



Translate your site

Let's translate `docs/intro.md` to French.

Configure i18n

Modify `docusaurus.config.js` to add support for the `fr` locale:

`docusaurus.config.js`

```
module.exports = {  
  i18n: {  
    defaultLocale: 'en',  
    locales: ['en', 'fr'],  
  },  
};
```

Translate a doc

Copy the `docs/intro.md` file to the `i18n/fr` folder:

```
mkdir -p i18n/fr/docusaurus-plugin-content-docs/current/  
  
cp docs/intro.md i18n/fr/docusaurus-plugin-content-docs/current/intro.md
```

Translate `i18n/fr/docusaurus-plugin-content-docs/current/intro.md` in French.

Start your localized site

Start your site on the French locale:

```
npm run start -- --locale fr
```

Your localized site is accessible at `http://localhost:3000/fr/` and the `Getting Started` page is translated.



In development, you can only use one locale at a same time.

Add a Locale Dropdown

To navigate seamlessly across languages, add a locale dropdown.

Modify the `docusaurus.config.js` file:

`docusaurus.config.js`

```
module.exports = {
  themeConfig: {
    navbar: {
      items: [
        {
          type: 'localeDropdown',
        },
      ],
    },
  },
};
```

The locale dropdown now appears in your navbar:

A🇫 Français ▼

English

Français

Build your localized site

Build your site for a specific locale:

```
npm run build -- --locale fr
```

Or build your site to include all the locales at once:

```
npm run build
```

