

Remixの紹介

Press Space for next page →



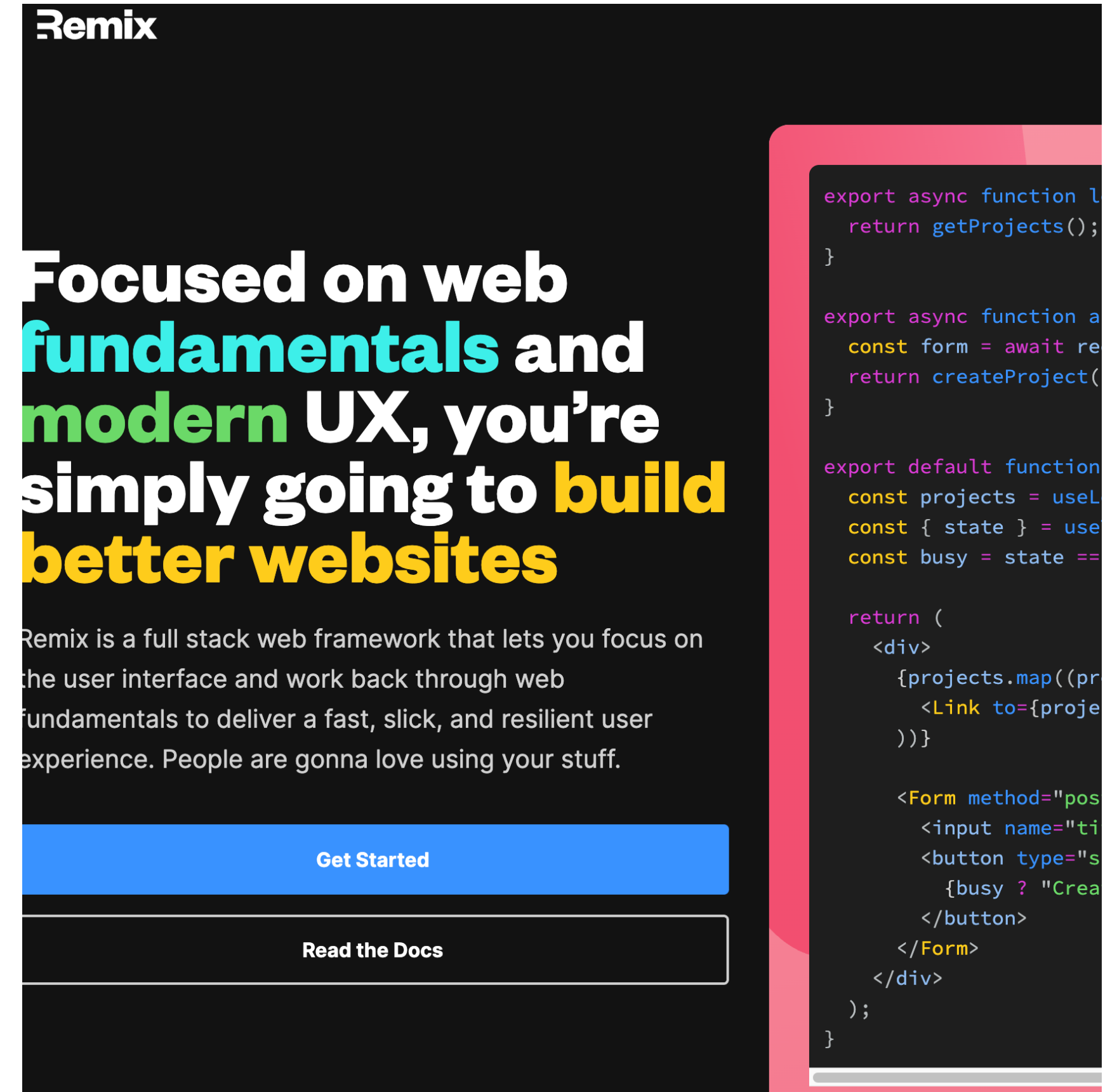


自己紹介

- 📝 飯野陽平 (wheatandcat)
- 🏢 フリーランスエンジニア (シェアフル株式会社CTO)
- 💻 Blog: <https://www.wheatandcat.me/>
- 📖 Booth: <https://wheatandcat.booth.pm/>
- 🛠️ 今までに作ったもの
 - memoir
 - ペペロミア
 - Atomic Design Check List

Remixとは？

2021年11月にOSSとして公開されたReactベースの新しいフレームワーク。



The image shows a promotional banner for the Remix web framework. It features a dark background with the Remix logo in the top left. The main headline is 'Focused on web fundamentals and modern UX, you're simply going to build better websites', with 'fundamentals' in cyan, 'modern' in green, and 'better websites' in yellow. Below the headline, a paragraph describes Remix as a full stack web framework that focuses on the user interface and web fundamentals to deliver a fast, slick, and resilient user experience. At the bottom, there are two buttons: 'Get Started' in blue and 'Read the Docs' in white with a black border. On the right side, there is a pink vertical bar containing a code snippet in a dark-themed editor.

Remix

Focused on web fundamentals and modern UX, you're simply going to build better websites

Remix is a full stack web framework that lets you focus on the user interface and work back through web fundamentals to deliver a fast, slick, and resilient user experience. People are gonna love using your stuff.

[Get Started](#)

[Read the Docs](#)

```
export async function loader() {
  return getProjects();
}

export async function action({ request }) {
  const form = await request.formData();
  return createProject({
    name: form.get('name'),
    description: form.get('description')
  });
}

export default function App() {
  const projects = useLoaderData();
  const { state } = useFormState();
  const busy = state.isLoading;

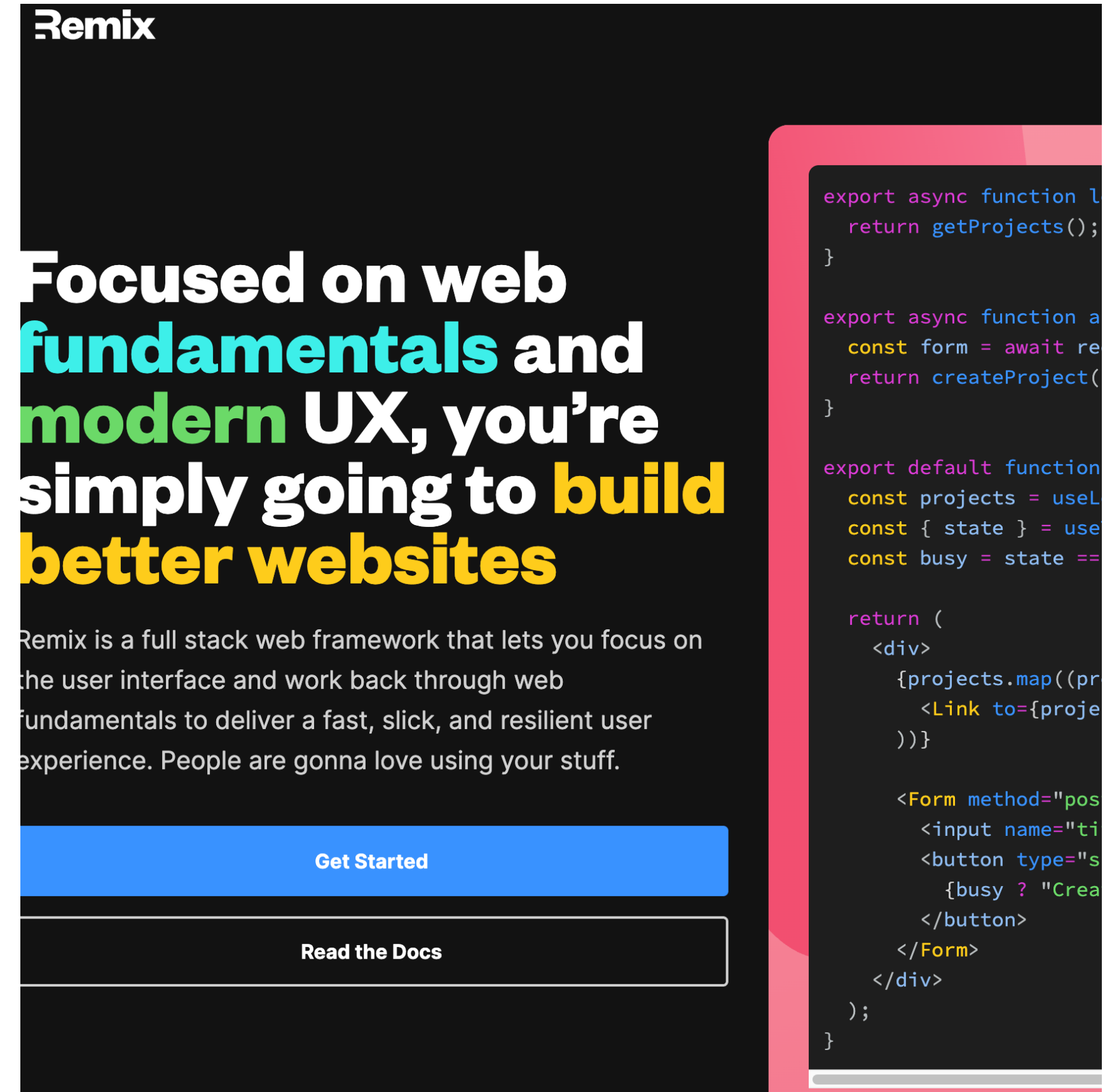
  return (
    <div>
      {projects.map((project) => (
        <Link to={project.slug} />
      ))}

      <Form method="post">
        <input name="title" type="text" />
        <button type="submit">
          {busy ? "Creating..." : "Create Project"}
        </button>
      </Form>
    </div>
  );
}
```

Remixとは？

2021年11月にOSSとして公開されたReactベースの新しいフレームワーク。

ReactRouterの作者を始めとして有名なエンジニアが参加しており、フロントエンド界隈で今話題になっているプロジェクト。

The image shows the Remix website landing page. It has a dark background with the Remix logo in the top left. The main headline is 'Focused on web fundamentals and modern UX, you're simply going to build better websites', with 'fundamentals' in cyan, 'modern' in green, and 'better websites' in yellow. Below the headline is a paragraph: 'Remix is a full stack web framework that lets you focus on the user interface and work back through web fundamentals to deliver a fast, slick, and resilient user experience. People are gonna love using your stuff.' At the bottom are two buttons: 'Get Started' in blue and 'Read the Docs' in white with a black border. On the right side, there is a pink vertical bar and a code editor showing JavaScript and JSX code for a projects list and a create project form.

Remix

Focused on web fundamentals and modern UX, you're simply going to build better websites

Remix is a full stack web framework that lets you focus on the user interface and work back through web fundamentals to deliver a fast, slick, and resilient user experience. People are gonna love using your stuff.

[Get Started](#)

[Read the Docs](#)

```
export async function load() {
  return getProjects();
}

export async function action({ request }) {
  const form = await request.formData();
  return createProject({
    title: form.get('title'),
    description: form.get('description')
  });
}

export default function App() {
  const { projects, isLoading } = useLoaderData();
  const { state, setState } = useState({
    title: '',
    description: ''
  });
  const busy = state.title.length > 0;

  return (
    <div>
      {projects.map((project) => (
        <Link to={project.slug} >{project.title}</Link>
      ))}

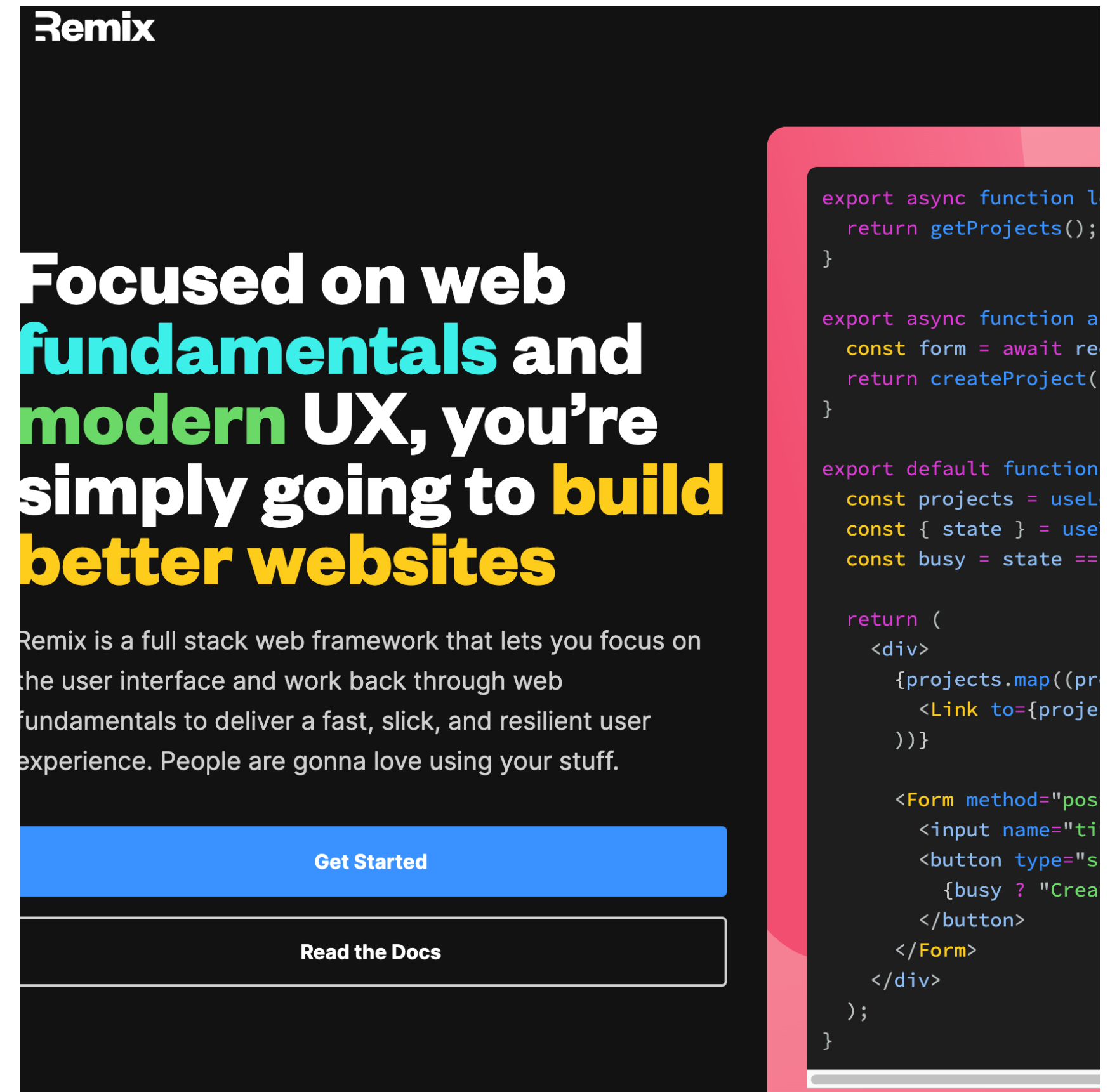
      <Form method="post">
        <input name="title" type="text" value={state.title} />
        <button type="submit">{busy ? "Creating..." : "Create Project"}</button>
      </Form>
    </div>
  );
}
```

Remixとは？

2021年11月にOSSとして公開されたReactベースの新しいフレームワーク。

ReactRouterの作者を始めとして有名なエンジニアが参加しており、フロントエンド界隈で今話題になっているプロジェクト。

現状だとWeb開発はNext.jsが利用される事が多いが、Next.jsとはアプローチが明確に異なる点が多く、比較されている記事もよく見かける。



The image shows a promotional banner for the Remix web framework. The background is dark with a pinkish-red gradient on the right side. The Remix logo is in the top left. The main headline is 'Focused on web fundamentals and modern UX, you're simply going to build better websites'. Below this is a paragraph describing Remix as a full stack web framework. At the bottom are two buttons: 'Get Started' and 'Read the Docs'. On the right, a code editor shows a snippet of JavaScript code for handling projects.

Remix

Focused on web fundamentals and modern UX, you're simply going to build better websites

Remix is a full stack web framework that lets you focus on the user interface and work back through web fundamentals to deliver a fast, slick, and resilient user experience. People are gonna love using your stuff.

[Get Started](#)

[Read the Docs](#)

```
export async function loader() {
  return getProjects();
}

export async function action({ request }) {
  const form = await request.formData();
  return createProject({
    name: form.get('name'),
  });
}

export default function App() {
  const [projects, setProjects] = useState([]);
  const [state, setState] = useState({});
  const [busy, setBusy] = useState(false);

  return (
    <div>
      {projects.map((project) => (
        <Link to={project.slug} />
      ))}

      <Form method="post">
        <input name="title" type="text" />
        <button type="submit">
          {busy ? "Creating..." : "Create Project"}
        </button>
      </Form>
    </div>
  );
}
```

Remixとは？

2021年11月にOSSとして公開されたReactベースの新しいフレームワーク。

ReactRouterの作者を始めとして有名なエンジニアが参加しており、フロントエンド界隈で今話題になっているプロジェクト。

現状だとWeb開発はNext.jsが利用される事が多いが、Next.jsとはアプローチが明確に異なる点が多く、比較されている記事もよく見かける。

ということで、今回はチュートリアルを見ながら基本的な部分を説明する。

Remix

Focused on web
fundamentals and
modern UX, you're
simply going to build
better websites

Remix is a full stack web framework that lets you focus on the user interface and work back through web fundamentals to deliver a fast, slick, and resilient user experience. People are gonna love using your stuff.

Get Started

Read the Docs

```
export async function load() {
  return getProjects();
}

export async function action({ request }) {
  const form = await request.formData();
  return createProject({
    title: form.get('title'),
    description: form.get('description')
  });
}

export default function App() {
  const projects = useLoaderData();
  const { state } = useState({});
  const busy = state.loading;

  return (
    <div>
      {projects.map((project) => (
        <Link to={project.slug} />
      ))}

      <Form method="post">
        <input name="title" type="text" />
        <button type="submit">
          {busy ? "Creating..." : "Create Project"}
        </button>
      </Form>
    </div>
  );
}
```

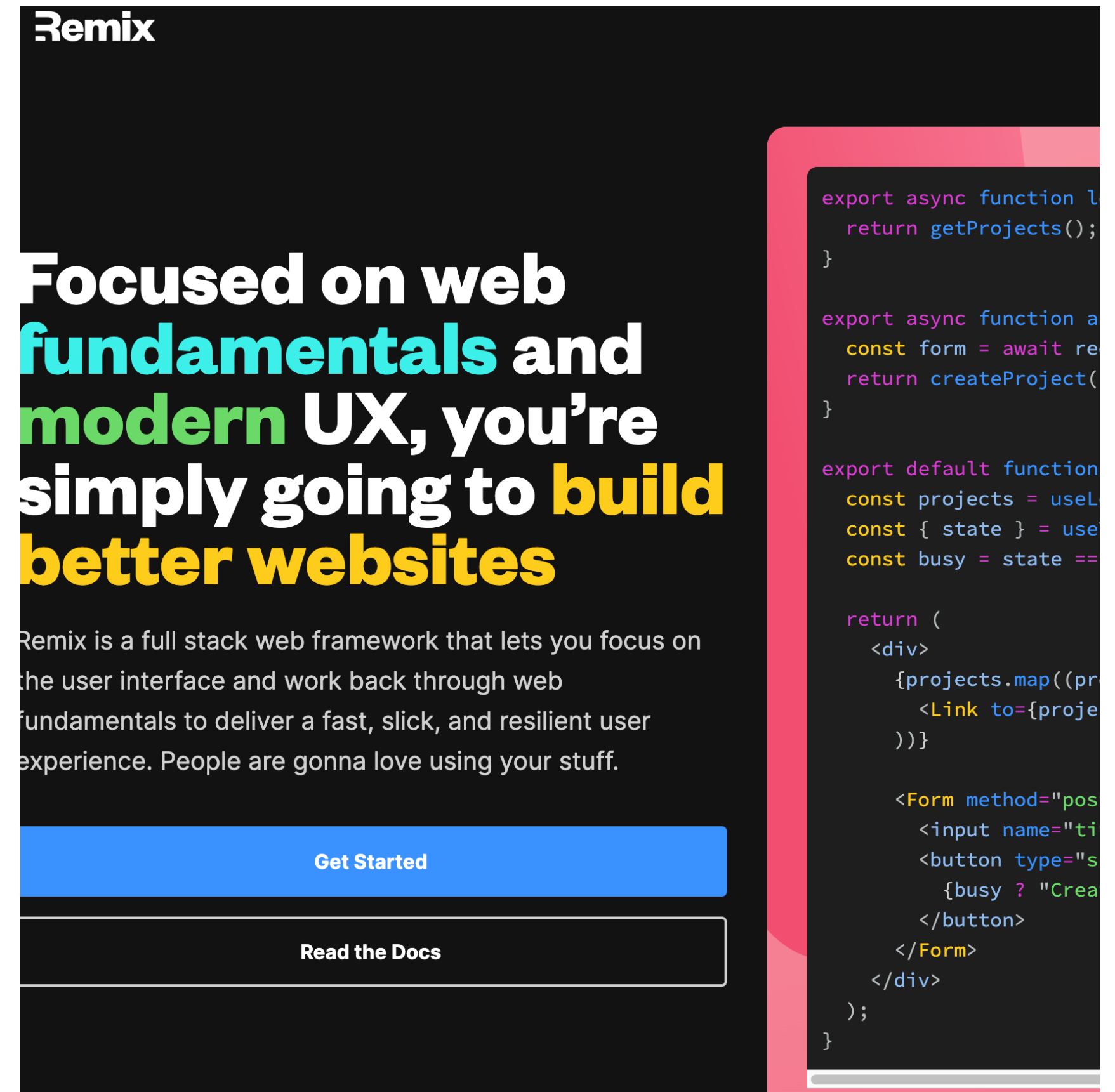
Remixとは？

2021年11月にOSSとして公開されたReactベースの新しいフレームワーク。

ReactRouterの作者を始めとして有名なエンジニアが参加しており、フロントエンド界隈で今話題になっているプロジェクト。

現状だとWeb開発はNext.jsが利用される事が多いが、Next.jsとはアプローチが明確に異なる点が多く、比較されている記事もよく見かける。

ということで、今回はチュートリアルを見ながら基本的な部分を説明する。

The image shows a promotional banner for the Remix web framework. It features a dark background with the Remix logo in the top left. The main headline is 'Focused on web fundamentals and modern UX, you're simply going to build better websites', with 'fundamentals' in cyan, 'modern' in green, and 'better websites' in yellow. Below the headline is a paragraph describing Remix as a full-stack framework that focuses on the user interface and web fundamentals. At the bottom, there are two buttons: 'Get Started' in blue and 'Read the Docs' in white with a black border. On the right side, there is a pink vertical bar containing a code editor snippet with JavaScript and JSX code for a project listing and a form.

Remix

Focused on web fundamentals and modern UX, you're simply going to build better websites

Remix is a full stack web framework that lets you focus on the user interface and work back through web fundamentals to deliver a fast, slick, and resilient user experience. People are gonna love using your stuff.

[Get Started](#)

[Read the Docs](#)

```
export async function loader() {
  return getProjects();
}

export async function action({ request }) {
  const form = await request.formData();
  return createProject({
    title: form.get('title'),
    description: form.get('description')
  });
}

export default function App() {
  const projects = useLoaderData();
  const { state } = useState({});
  const busy = state.loading;

  return (
    <div>
      {projects.map((project) => (
        <Link to={project.slug} />
      ))}

      <Form method="post">
        <input name="title" type="text" />
        <button type="submit">
          {busy ? "Creating..." : "Create Project"}
        </button>
      </Form>
    </div>
  );
}
```

Remixのポリシー

Remixのwebページには以下のようなポリシーが挙げられている

1. Embrace the server/client model, including separation of source code from content/data.
2. Work with, not against, the foundations of the web: Browsers, HTTP, and HTML. It's always been good and it's gotten really good in the last few years.
3. Use JavaScript to augment the user experience by emulating browser behavior.
4. Don't over-abstract the underlying technologies

[Learn More](#)

チュートリアルをやってみる

以下からチュートリアルを行う事が可能

<https://remix.run/docs/en/v1/tutorials/jokes>

特徴

- パフォーマンスのアプローチの違い
- NestedRoutingのサポート
- JavaScriptをONにしなくても動作可能
- HTMLに準拠したコーディングが可能
- シンプルなステート管理

パフォーマンスのアプローチの違い

Next.jsではパフォーマンス向上のためSSGやISR^[1]のサポートし、静的サイトとして運用することで、CDNにキャッシュが可能になり高速化を行ってきた

[Learn More](#)

[Next.jsのISRで動的コンテンツをキャッシュするときの戦略](#) 

[Cloudflare Workers documentation](#) 

パフォーマンスのアプローチの違い

Next.jsではパフォーマンス向上のためSSGやISR^[1]のサポートし、静的サイトとして運用することで、CDNにキャッシュが可能になり高速化を行ってきた

Remixで以下でパフォーマンス向上を行っている

- 1.Cloudflare Workers^[2]などのエッジコンピューティングを積極的に活用
- 2.各Router毎にCache Controlを行う
- 3.ブラウザ上から通信で取得するファイルを極限まで減らしている

Learn More

Next.jsのISRで動的コンテンツをキャッシュするときの戦略 

Cloudflare Workers documentation 

パフォーマンスのアプローチの違い

Next.jsではパフォーマンス向上のためSSGやISR^[1]のサポートし、静的サイトとして運用することで、CDNにキャッシュが可能になり高速化を行ってきた

Remixで以下でパフォーマンス向上を行っている

- 1.Cloudflare Workers^[2]などのエッジコンピューティングを積極的に活用
- 2.各Router毎にCache Controlを行う
- 3.ブラウザ上から通信で取得するファイルを極限まで減らしている

上記を正しく動作させることでSSGやISRに遜色ないパフォーマンスが出せる（らしい）

[Learn More](#)

[Next.jsのISRで動的コンテンツをキャッシュするときの戦略](#) 

[Cloudflare Workers documentation](#) 

まとめ

公開されたばかりののフレームワークで、プロダクトで使えるかまでは未検証ですが、ちょっとした開発では使ってみたいと思えるくらいには作り込まれている感じでした。

web開発はNext.jsの一強の状態だったところに、方向性の違うRemixが出てきて、また盛り上がっている感じなので、今後も期待かなと思います。

ご清聴ありがとうございました