

Gatsby is built on top of React. So anything that is possible with a React form is possible in Gatsby. Additional details about how to create React forms can be found in the [React forms documentation](#) (which happens to be built with Gatsby!)

Start with the following page.

```
import React from "react"

export default function Home() {
  return <div>Hello world!</div>
}
```

This Gatsby page is a React component. When you want to create a form, you need to store the state of the form - what the user has entered. Convert your function (stateless) component to a class (stateful) component.

```
import React from "react"

export default class IndexPage extends React.Component {
  render() {
    return <div>Hello world!</div>
  }
}
```

Now that you have created a class component, you can add `state` to the component.

```
import React from "react"

export default class IndexPage extends React.Component {
  state = {
    firstName: "",
    lastName: "",
  }

  render() {
    return <div>Hello world!</div>
  }
}
```

And now you can add a few input fields:

```
import React from "react"

export default class IndexPage extends React.Component {
  state = {
    firstName: "",
    lastName: "",
  }

  render() {
    return (
```

```

    <form>
      <label>
        First name
        <input type="text" name="firstName" />
      </label>
      <label>
        Last name
        <input type="text" name="lastName" />
      </label>
      <button type="submit">Submit</button>
    </form>
  )
}
}

```

When a user types into an input box, the state should update. Add an `onChange` prop to update state and add a `value` prop to keep the input up to date with the new state:

```

import React from "react"

export default class IndexPage extends React.Component {
  state = {
    firstName: "",
    lastName: "",
  }

  handleInputChange = event => {
    const target = event.target
    const value = target.value
    const name = target.name

    this.setState({
      [name]: value,
    })
  }

  render() {
    return (
      <form>
        <label>
          First name
          <input
            type="text"
            name="firstName"
            value={this.state.firstName}
            onChange={this.handleInputChange}
          />
        </label>
        <label>
          Last name
          <input

```

```

        type="text"
        name="lastName"
        value={this.state.lastName}
        onChange={this.handleInputChange}
      />
    </label>
    <button type="submit">Submit</button>
  </form>
)
}
}

```

Now that your inputs are working, you want something to happen when you submit the form. Add `onSubmit` props to the form element and add `handleSubmit` to show an alert when the user submits the form:

```

import React from "react"

export default class IndexPage extends React.Component {
  state = {
    firstName: "",
    lastName: "",
  }

  handleInputChange = event => {
    const target = event.target
    const value = target.value
    const name = target.name

    this.setState({
      [name]: value,
    })
  }

  handleSubmit = event => {
    event.preventDefault()
    alert(`Welcome ${this.state.firstName} ${this.state.lastName}!`)
  }

  render() {
    return (
      <form onSubmit={this.handleSubmit}>
        <label>
          First name
          <input
            type="text"
            name="firstName"
            value={this.state.firstName}
            onChange={this.handleInputChange}
          />
        </label>
        <label>

```

```
      Last name
      <input
        type="text"
        name="lastName"
        value={this.state.lastName}
        onChange={this.handleInputChange}
      />
    </label>
    <button type="submit">Submit</button>
  </form>
)
}
}
```

This form isn't doing anything besides showing the user information that they just entered. At this point, you may want to move this form to a component, send the form state to a Function (see the [form example](#)), or add robust validation. You can also use fantastic React form libraries like [Formik](#) or [Final Form](#) to speed up your development process.

All of this is possible and more by leveraging the power of Gatsby and the React ecosystem!