

React Props

Table of Contents:

- React Props
- Props destructuring
- Pass an object as a prop
- spread operator in prop

React Props

In React, components communicate by passing data through props (short for "properties"). Props allow parent components to send data to child component

- Props are **read-only** and **cannot be modified** inside the child component.
- They are passed as **attributes** in JSX.
- They help make components **reusable and dynamic**.

```
function Greeting(props) {  
  return <h1>Hello, {props.name}!</h1>;  
}  
  
function App() {  
  return <Greeting name="Alice" />;  
}
```

Props Destructuring

- Instead of using `props.name`, you can **destructure** props directly in function parameters.

```
function Greeting({ name }) {  
  return <h1>Hello, {name}!</h1>;  
}
```

Default Props

- You can set **default values** for props in case they are not provided.

```
function Greeting({ name = "Guest" }) {  
  return <h1>Hello, {name}!</h1>;  
}
```

Passing an Object as a Prop

- You can pass an **entire object** as a prop instead of individual properties

```
function UserInfo({ user }) {  
  return <h1>{user.name} is {user.age} years old.</h1>;  
}  
  
const user = { name: "Alice", age: 25 };  
function App() {  
  return <UserInfo user={user} />;  
}
```

Spread Operator in Props

- The **spread operator (...)** allows passing all object properties as props

```
function UserInfo({ name, age }) {  
  return <h1>{name} is {age} years old.</h1>;  
}  
  
const user = { name: "Alice", age: 25 };  
function App() {  
  return <UserInfo {...user} />;  
}
```

- This avoids manually passing each property (`name={user.name}, age={user.age}`).

The `children` prop

When you nest content inside a JSX tag, the parent component will receive that content in a prop called `children`

```
import Avatar from './Avatar.js';

function Card({ children }) {
  return (
    <div className="card">
      {children}
    </div>
  );
}

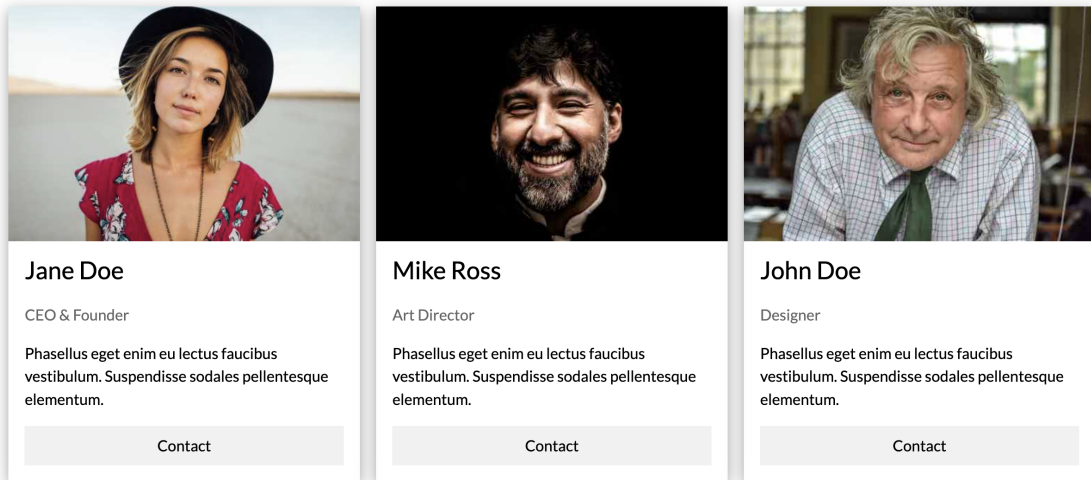
export default function Profile() {
  return (
    <Card>
      <Avatar
        size={100}
        person={{
          name: 'Katsuko Saruhashi',
          imageId: 'Yfe0qp2'
        }}
      />
    </Card>
  );
}
```

You use the `children` prop to create HOC (Higher-Order Components) e.g. Modal, Tooltip, etc.

Assignments

Use Props to pass User Display Data:

1. Profiles Display



2. What do we offer



The Latest and Greatest

Kits auto-update to the newest version of Font Awesome, so you can stop managing files and pushing code.



Auto-Accessibility & More

Automate your icons' accessibility, control load timing, ensure Font Awesome Version 4 compatibility, and more.



Super Simple Subsetting

Speed up your site with a super-lean, ultra-fast version of Font Awesome! Choose just the icons you need, and your Kit serves them up as a subset, making it faster than ever.



Powered by Cloudflare

Font Awesome Kits are served using Cloudflare, so you know they're secure, private, fast, and reliable.

3. Build this section with red boxes as separate components.

