React Notes

Laiba Imran

Components

- Components are one of the core concepts of React. They are the foundation upon which you build user interfaces (UI).
- React lets you combine your markup, CSS, and JavaScript into custom “components”, reusable UI elements for your app.

Components

App.js method 1

```
function Profile() {
  return (
    <img
      src="https://i.imgur.com/MK3eW3Am.jpg"
      alt="Katherine Johnson"
    />
  )
}

export default Profile;
```

Click to see the results of above code

https://codesandbox.io/p/sandbox/react-dev-forked-wsvgtx

Components

App.js method 2

```
export default function Profile() {
  return (
    <img
      src="https://i.imgur.com/MK3eW3Am.jpg"
      alt="Katherine Johnson"
    />
  )
}
```

Markups

- The component returns an <img /> tag with src and alt attributes. <img /> is written like HTML, but it is actually JavaScript under the hood! This syntax is called JSX, and it lets you embed markup inside JavaScript.

- Return statements can be written all on one line, as in this component:

```
return <img src="https://i.i.jpg" alt="Katherine Johnson" />;
```

- But if your markup isn't all on the same line as the return keyword, you must wrap it in a pair of parentheses:

```
return (
    <div>
        <img src="https://i.i.jpg" alt="Katherine Johnson" />
    </div>
);
```

Using a Component

App.js

```
function Profile() {
  return (
    <img
      src="https://i.imgur.com/MK3eW3As.jpg"
      alt="Katherine Johnson"
    />
  );
}

export default function Gallery() {
  return (
    <section>
      <h1>Amazing scientists</h1>
      <Profile />
      <Profile />
      <Profile />
    </section>
  );
}
```

Click to see the results of above code

https://codesandbox.io/p/sandbox/xgy8t9?file=%2Fsrc%2FApp.js%3A1%2C1-20%2C1

Exporting and Importing Components

- The magic of components lies in their reusability: you can create components that are composed of other components.
- But as you nest more and more components, it often makes sense to start splitting them into different files.
- This lets you keep your files easy to scan and reuse components in more places.

Exporting and Importing Components

Gallery.js

```
function Profile() {
  return (
    <img
      src="https://i.imgur.com/QIrZWGIs.jpg"
      alt="Alan L. Hart"
    />
  );
}

export default function Gallery() {
  return (
    <section>
      <h1>Amazing scientists</h1>
      <Profile />
      <Profile />
      <Profile />
    </section>
  );
}
```

App.js

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

export default function App() {
  return (
    <Gallery />
  );
}
```

Click to see the results of above code

https://codesandbox.io/p/sandbox/nfmfwy?file=%2Fsrc%2FApp.js%3A1%2C1-7%2C2

Exporting and Importing Multiple Components

Gallery.js

```
export function Profile() {
  return (
    <img
      src="https://i.imgur.com/QIrZWGIs.jpg"
      alt="Alan L. Hart"
    />
  );
}

export default function Gallery() {
  return (
    <section>
      <h1>Amazing scientists</h1>
      <Profile />
      <Profile />
      <Profile />
    </section>
  );
}
```

App.js

```
import Gallery from './Gallery.js';
import { Profile } from './Gallery.js';

export default function App() {
  return (
    <Profile />
  );
}
```

Click to see the results of above code

https://codesandbox.io/p/sandbox/5qcqj6?file=%2Fsrc%2FGallery.js%3A1%2C1-19%2C2

Writing JSX

- JSX is a syntax extension for JavaScript that lets you write HTML-like markup inside a JavaScript file.

- Although there are other ways to write components, most React developers prefer the conciseness of JSX, and most codebases use it.

Converting HTML to JSX

App.js

```
export default function TodoList() {
  return (
    // This doesn't quite work!
    <h1>Hedy Lamarr's Todos</h1>
    <img
      src="https://i.imgur.com/yXOvdOS
s.jpg"
      alt="Hedy Lamarr"
      class="photo"
    >
    <ul>
      <li>Invent new traffic lights
      <li>Rehearse a movie scene
      <li>Improve the spectrum
    </ul>
  );
}
```

Simply adding HTML to your component will not work. This is because JSX is stricter and has a few more rules than HTML!

TypeError

```
Cannot assign to read only property 'message' of
object 'SyntaxError: /src/App.js: Adjacent JSX
elements must be wrapped in an enclosing tag. Did you
want a JSX fragment <>...</>? (5:4)

  3 |     // This doesn't quite work!
  4 |     <h1>Hedy Lamarr's Todos</h1>
> 5 |     <img
    |     ^
```

Click to see the results of above code

https://codesandbox.io/p/sandbox/9m6zmf?file=%2Fsrc%2FApp.js%3A1%2C1-16%2C2

Converting to JSX

1. Return a single root element

App.js

```
export default function TodoList() {
  return (
   <div>
    <h1>Hedy Lamarr's Todos</h1>
    <img
      src="https://i.imgur.com/yXOvdOS
s.jpg"
      alt="Hedy Lamarr"
      class="photo"
    >
    <ul>
      <li>Invent new traffic lights
      <li>Rehearse a movie scene
      <li>Improve the spectrum
    </ul>
   </div>
  );
}
```

To return multiple elements from a component, wrap them with a single parent tag.

For example, you can use a :

1. Return a single root element

App.js

```
export default function TodoList() {
  return (
    <>
      <h1>Hedy Lamarr's Todos</h1>
      <img
        src="https://i.imgur.com/yXOvdOS
s.jpg"
        alt="Hedy Lamarr"
        class="photo"
      >
      <ul>
        <li>Invent new traffic lights
        <li>Rehearse a movie scene
        <li>Improve the spectrum
      </ul>
    </>
  );
}
```

If you don’t want to add an extra to your markup, you can write <> and </> instead:

2. Close All Tags

App.js

```
export default function TodoList() {
  return (
    <>
      <h1>Hedy Lamarr's Todos</h1>
      <img
        src="https://i.imgur.com/yXOvdOSs.jpg"
        alt="Hedy Lamarr"
        class="photo"
      />
      <ul>
        <li>Invent new traffic lights </li>
        <li>Rehearse a movie scene </li>
        <li>Improve the spectrum </li>
      </ul>
    </>
  );
}
```

JSX requires tags to be explicitly closed: self-closing tags like <img> must become <img />, and wrapping tags like oranges must be written as oranges.

App.js

```
export default function TodoList() {
  return (
    <>
      <h1>Hedy Lamarr's Todos</h1>
      <img
        src="https://i.imgur.com/yXOvdOSs.jpg"
        alt="Hedy Lamarr"
        className="photo"
      />
      <ul>
        <li>Invent new traffic lights </li>
        <li>Rehearse a movie scene </li>
        <li>Improve the spectrum </li>
      </ul>
    </>
  );
}
```

3. camelCase most of the things!

For example, class is a reserved word, in React you write className instead, named after the corresponding DOM property:

Passing Props to a Component

- React components use props to communicate with each other.
- Every parent component can pass some information to its child components by giving them props.
- Props might remind you of HTML attributes, but you can pass any JavaScript value through them, including objects, arrays, and functions.
- In fact, props are the only argument to your component!

Passing props to Components

App.js method 1

```
import { getImageUrl } from './utils.js';

function Avatar({ size, person}) {
  return (
    <img
      className="avatar"
      src={getImageUrl(person)}
      alt={person.name}
      width={size}
    /> );}

export default function Profile() {
  return (
    <div>
      <Avatar
        size={100}
        person={{
          name: 'Katsuko Saruhashi',
          imageId: 'YfeOqp2'
        }} />
    </div> );}
```

Syntax 1:

function Avatar({ person, size })

Click to see the results of above code

https://codesandbox.io/p/sandbox/react-dev-forked-lfvrwd?file=%2Fsrc%2FApp.js%3A1%2C1-26%2C2

Passing props to Components

App.js method 2

```
import { getImageUrl } from './utils.js';

function Avatar(props) {
  return (
    <img
      className="avatar"
      src={getImageUrl(props.person)}
      alt={props.person.name}
      width={props.size}
    />);}

export default function Profile() {
  return (
    <div>
      <Avatar
        size={100}
        person={{
          name: 'Katsuko Saruhashi',
          imageId: 'YfeOqp2'
        }} />
    </div> );}
```

Syntax 2:

function Avatar(props)

Click to see the results of above code

https://codesandbox.io/p/sandbox/react-dev-forked-lfvrwd?file=%2Fsrc%2FApp.js%3A1%2C1-26%2C2

Default values of props

App.js

```
import { getImageUrl } from './utils.js';

function Avatar({ size=100, person}) {
  return (
    <img
      className="avatar"
      src={getImageUrl(person)}
      alt={person.name}
      width={size}
    /> );}

export default function Profile() {
  return (
    <div>
      <Avatar
        size={100}
        person={{
          name: 'Katsuko Saruhashi',
          imageId: 'YfeOqp2'
        }} />
    </div> );}
```

You can pass default values to props as well

Click to see the results of above code

https://codesandbox.io/p/sandbox/react-dev-forked-lfvrwd?file=%2Fsrc%2FApp.js%3A1%2C1-26%2C2

Resources

Go over https://react.dev/learn/describing-the-ui