Your First Component

Components are one of the core concepts of React. They are the foundation upon which you build user interfaces (UI), which makes them the perfect place to start your React journey!

You will learn

- What a component is
- What role components play in a React application
- How to write your first React component

Components: UI building blocks

On the Web, HTML lets us create rich structured documents with its built-in set of tags like `<h1>` and `<li>`:

```
<article>
  <h1>My First Component</h1>
  <ol>
    <li>Components: UI Building Blocks</li>
    <li>Defining a Component</li>
    <li>Using a Component</li>
  </ol>
</article>
```

This markup represents this article `<article>`, its heading `<h1>`, and an (abbreviated) table of contents as an ordered list `<ol>`. Markup like this, combined

React lets you combine your markup, CSS, and JavaScript into custom "components," **reusable UI elements for your app.** The table of contents code you saw above could be turned into a `<TableOfContents />` component you could render on every page. Under the hood, it still uses the same HTML tags like `<article>`, `<h1>`, etc.

Just like with HTML tags, you can compose, order and nest components to design whole pages. For example, the documentation page you're reading is made out of React components:

```
<PageLayout>
  <NavigationHeader>
    <SearchBar />
    <Link to="/docs">Docs</Link>
  </NavigationHeader>
  <Sidebar />
  <PageContent>
    <TableOfContents />
    <DocumentationText />
  </PageContent>
</PageLayout>
```

As your project grows, you will notice that many of your designs can be composed by reusing components you already wrote, speeding up your development. Our table of contents above could be added to any screen with `<TableOfContents />`! You can even jumpstart your project with the thousands of components shared by the React open source community like Chakra UI and Material UI.

Defining a component

Traditionally when creating web pages, web developers marked up their content and then added interaction by sprinkling on some JavaScript. This worked great when interaction was a nice-to-have on the web. Now it is expected for many sites and all that looks like (you can edit the example below):

App.js ⌄ Download Reset Fork

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

And here's how to build a component:

Step 1: Export the component

The `export default` prefix is a standard JavaScript syntax (not specific to React). It lets you mark the main function in a file so that you can later import it from other files. (More on importing in Importing and Exporting Components!)

Step 2: Define the function

With `function Profile() { }` you define a JavaScript function with the name `Profile`.

Gotcha

React components are regular JavaScript functions, but **their names must start with a capital letter** or they won't work!

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.imgur.com/MK3eW3As.jpg" alt="Katherine Johnson" />;
```

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

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

Gotcha

Without parentheses, any code on the lines after `return` will be ignored!

Using a component

Now that you've defined your `Profile` component, you can nest it inside other components. For example, you can export a `Gallery` component that uses multiple `Profile` components:

App.js Download Reset Fork

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

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

Show more

What the browser sees

Notice the difference in casing:

- `<section>` is lowercase, so React knows we refer to an HTML tag.
- `<Profile />` starts with a capital `P`, so React knows that we want to use our component called `Profile`.

And `Profile` contains even more HTML: `<img />`. In the end, this is what the browser sees:

```
<section>
  <h1>Amazing scientists</h1>
  <img src="https://i.imgur.com/MK3eW3As.jpg" alt="Katherine Johnson" />
  <img src="https://i.imgur.com/MK3eW3As.jpg" alt="Katherine Johnson" />
  <img src="https://i.imgur.com/MK3eW3As.jpg" alt="Katherine Johnson" />
</section>
```

Nesting and organizing components

related to each other. If this file gets crowded, you can always move `Profile` to a separate file. You will learn how to do this shortly on the page about imports.

Because the `Profile` components are rendered inside `Gallery`—even several times!—we can say that `Gallery` is a **parent component,** rendering each `Profile` as a “child”. This is part of the magic of React: you can define a component once, and then use it in as many places and as many times as you like.

DEEP DIVE

Components All the Way Down

Show Details

Recap

You’ve just gotten your first taste of React! Let’s recap some key points.

- React lets you create components, **reusable UI elements for your app.**
- In a React app, every piece of UI is a component.
- React components are regular JavaScript functions except:
  1. Their names always begin with a capital letter.
  2. They return JSX markup.

Try out some challenges

1. Export the component 2. Fix the return statement 3. Spot the mistake 4. Your

This sandbox doesn't work because the root component is not exported:

App.js

Download Reset Fork

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

Try to fix it yourself before looking at the solution!

Show solution

Next Challenge

PREVIOUS
Describing the UI

NEXT
Importing and Exporting Components

FACEBOOK

Open Source

React Docs

Learn React

Quick Start

Installation

Describing the UI

Adding Interactivity

Managing State

Escape Hatches

API Reference

React APIs

React DOM APIs

Community

Code of Conduct

Acknowledgements

Meet the Team

More

React Native

Privacy

Terms