

Lesson 2 Exercises - Components and Props

Recommended: create a copy of react-basics.html to do these exercises.

2a. Create an `<App>` component that returns a paragraph with the text:

“Hello, welcome to my website” inside.

Then, render this `<App>`.

Hello, welcome to my website

2b. Continuing from 2a, in the `<App>` component, add 2 buttons after the paragraph: “Login” and “Sign up”

- Use a Fragment to group them together
- Use `()` to write the JSX on multiple lines

Hello, welcome to my website

Login Sign up

Solutions in description

2c. Continuing from 2b, in the `<App>` component, add 2 text boxes above the buttons with the placeholders “Email” and “Password”.

- Lay out the textboxes on separate lines using `<div>`
- Try adding the attribute `type="password"` to the password textbox. This will cover up the text as you type

Hello, welcome to my website

Email
Password
Login Sign up

2d. Continuing from 2c, create a new component called `LoginForm`, and move the textboxes and buttons (all the login-related code) into this component. Then, inside `<App>`, use `<LoginForm>` to create the textboxes and buttons.

Solutions in description

Challenge Exercises

Note: for these next exercises, start again with an empty React website.

- 2e. Let's say we're creating an e-commerce app, and one of the products is socks. Create an `<App>` component and display the details on the right onto the website:

Cotton socks

Price: \$10.90

Add to Cart

- 2f. Display 2 more products on the website: tennis balls and a t-shirt.

Instead of duplicating the code, create a `<ProductDetails>` component, and use props to make it reusable. Then, display 3 `<ProductDetails>` on the website for the socks, tennis balls, and t-shirt.

Tennis balls

Price: \$6.00

Add to Cart

Plain T-Shirt

Price: \$7.99

Add to Cart

- 2g. Continuing from 2f, some products can have a discount. Add a prop to `<ProductDetails>` called `discountPrice`. If `discountPrice` exists, display it under the original price.

- Give the socks product a discount price of \$5.45

Cotton socks

Price: \$10.90

Discount price: \$5.45

Add to Cart

- 2h. If a product has a `discountPrice` also cross out the original price.

- To cross out text, surround it with the element: `...`
- Don't cross out the original price if there's no discount. To check if `discountPrice` does not exist, use `!discountPrice`
- There are several ways to do this exercise. As long as you get the final result, you're good!

Cotton socks

~~Price: \$10.90~~

Discount price: \$5.45

Add to Cart

2i. JavaScript has a feature called the ternary operator:

```
const result = value1 ? value2 : value3;
```

[react-course](#) / [1-exercise-solutions](#) / [lesson-02](#) / README.md

↑ Top

Preview Code Blame

Raw    

2j. We'll add an image at the top of each Product. In your browser, go to `supersimple.dev/images/___` (replace `___` with `cotton-socks.png`, `tennis-balls.png`, and `plain-t-shirt.png`). Download the images.

- In `<ProductDetails>` add an `` at the top, and add a prop called `imageSrc`. Give the `` the attribute `src={imageSrc}` (this sets the `src` attribute to the value of the `imageSrc` prop).
- Then in `<App>`, give each `<ProductDetails>` a prop `imageSrc="___"` (replace `___` with the file name of the image, like `cotton-socks.png`)