



## MODULE 1: FOUNDATIONS (GETTING STARTED)

---

### LESSON 3: Your First Component

#### Learning Objective

By the end of this lesson, you will understand what React components are and build your first reusable component.

---

#### 3.1 What is a Component?

Imagine building a house with LEGO blocks. Each LEGO piece is a component.

You can:

- Use the same piece in multiple places
- Combine pieces to make something bigger
- Replace one piece without breaking everything

In web development, a component is a reusable piece of code that controls a part of your webpage.

Examples of Components:

- Header (appears on every page)
  - Product Card (appears many times on one page)
  - Footer (appears on every page)
  - Search Bar (can be placed anywhere)
- 

## 3.2 Our First Component: Header

Let's create a header component that will appear on every page of CeylonMart.

Step 1: Create a new folder called `components` inside your `app` folder

- In VS Code, right-click on the `app` folder
- Select "New Folder"
- Name it: `components`

Step 2: Create a new file called `Header.js` inside the `components` folder

- Right-click on the `components` folder
- Select "New File"
- Name it: `Header.js`

Step 3: Write your first component

In `app/components/Header.js`, add this code:

```
javascript
```

```
// This is a React component - notice it starts with capital letter
```

```

function Header() {
  return (
    <header style={{
      backgroundColor: '#1e40af',
      color: 'white',
      padding: '1rem',
      display: 'flex',
      justifyContent: 'space-between',
      alignItems: 'center'
    }}>
      <div style={{ fontSize: '1.5rem', fontWeight: 'bold' }}>
        🏠 CeylonMart
      </div>
      <nav>
        <a href="#" style={{ color: 'white', margin: '0 1rem', textDecoration: 'none' }}>Home</a>
        <a href="#" style={{ color: 'white', margin: '0 1rem', textDecoration: 'none' }}>Products</a>
        <a href="#" style={{ color: 'white', margin: '0 1rem', textDecoration: 'none' }}>Cart</a>
      </nav>
    </header>
  );
}

```

*// This line makes the component available to other files*

```
export default Header;
```

What's happening here?

- `function Header()` defines our component
  - The `return` statement contains what gets displayed (JSX - JavaScript XML)
  - `style={{ }}` adds CSS styles directly
  - `export default Header` allows other files to use this component
-

### 3.3 Using Your Component

Now let's use the Header component on our homepage.

Step 1: Open `app/page.js`

Step 2: Import and use your Header component:

```
javascript
// Import the Header component
import Header from './components/Header';

export default function Home() {
  return (
    <div>
      <Header />
      <main style={{ padding: '2rem' }}>
        <h1>Welcome to CeylonMart</h1>
        <p>Your one-stop shop for authentic Sri Lankan products.</p>
      </main>
    </div>
  );
}
```

Step 3: Save and check your browser



You should now see a blue header with "CeylonMart" and navigation links at the top of your page!

You've just created and used your first React component!

---

### 3.4 Component Rules (Important!)

Remember these rules when creating components:

Rule	Correct 	Incorrect 
Component names must start with capital letter	<code>function Header()</code>	<code>function header()</code>
Components must return something	<code>return &lt;div&gt;Hello&lt;/div&gt;</code>	<code>return;</code> (nothing)
Always have one parent element	<code>&lt;div&gt;&lt;p&gt;Hi&lt;/p&gt;&lt;/div&gt;</code>	<code>&lt;p&gt;Hi&lt;/p&gt;&lt;p&gt;Bye&lt;/p&gt;</code> (two elements)
Export the component	<code>export default Header</code>	Forgetting to export

### 3.5 Creating a Footer Component

Let's practice by creating another component.

Step 1: Create `app/components/Footer.js`

javascript

```
function Footer() {  
  return (  
    <footer style={{  
      backgroundColor: '#1e293b',  
      color: 'white',
```

```
padding: '2rem',
marginTop: '2rem',
textAlign: 'center'
}}>
<p>© 2026 CeylonMart - Sri Lankan Quality Products</p>
<p style={{ marginTop: '0.5rem', color: '#94a3b8' }}>
  Colombo | Kandy | Galle | Jaffna
</p>
</footer>
);
}
```

```
export default Footer;
```

Step 2: Use it in `app/page.js`

```
javascript
```

```
import Header from './components/Header';
import Footer from './components/Footer';

export default function Home() {
  return (
    <div>
      <Header />
      <main style={{ padding: '2rem', minHeight: '60vh' }}>
        <h1>Welcome to CeylonMart</h1>
        <p>Your one-stop shop for authentic Sri Lankan products.</p>
      </main>
      <Footer />
    </div>
  );
}
```

---

## 3.6 JSX: JavaScript + HTML

JSX is what makes React special. It lets you write HTML-like code inside JavaScript.

What's happening behind the scenes:

When you write:

```
javascript
function Header() {
  return <h1>Hello</h1>;
}
```

React converts it to:

```
javascript
function Header() {
  return React.createElement('h1', null, 'Hello');
}
```

But you don't need to worry about that! Just write JSX, and React handles the rest.

JSX Rules:

1. Use `className` instead of `class` (because `class` is a JavaScript keyword)
2. javascript

```
// Correct
<div className="container">
```

```
// Wrong
```

3. `<div class="container">`
4. Self-closing tags must have a slash
5. javascript

*// Correct*

```

```

*// Wrong*

6. ``
7. JavaScript expressions go in `{}`
8. javascript

```
const name = "Ravindu";
```

9. `return <h1>Hello, {name}</h1>;`
- 

## Lesson 3 Summary

- Components are reusable pieces of code
  - Component names must start with a capital letter
  - Import components to use them in other files
  - JSX lets you write HTML in JavaScript
  - Components help organize your code
- 

## Practice Exercises

1. Create a ProductCard component that shows:

- Product image (use a placeholder for now)
  - Product name
  - Product price in Rupees (Rs. 1,500)
2. Create a Navigation component that has links to different categories: "Electronics", "Clothing", "Food", "Home"
  3. Challenge: Try adding a simple counter in a new component that shows how many items are in cart (hardcode the number for now)
- 

## Module 1 Quiz

1. What command creates a new Next.js app?
2. Where do you write your homepage content?
3. What's wrong with this component name: `function header()`?
4. What does `export default` do?
5. How do you include JavaScript inside JSX?

\*(Answers: 1-`npx create-next-app`, 2-`app/page.js`, 3-Must start with capital H, 4-Makes component available to other files, 5-Using curly braces `{}`)\*