

# React Component Composition

Now we'll discover how to use a React element in the same fashion as an HTML element, with an opening and closing tag:

```
const App = () => {  
  ...  
  
  return (  
    <div>  
      <h1>My Hacker Stories</h1>  
  
      <InputWithLabel  
        id="search"  
        value={searchTerm}  
        onChange={handleSearch}  
      >  
        Search  
      </InputWithLabel>  
  
      ...  
    </div>  
  );  
};
```

src/App.js

Instead of using the `label` prop from before, we inserted the text “Search:” between the component’s element’s tags. In the `InputWithLabel` component, you have access to this information via **React’s children** prop. Instead of using the `label` prop, use the `children` prop to render everything that has been passed down from above where you want it:

```
const InputWithLabel = ({  
  id,  
  value,  
  type = 'text',  
  onChange,  
  
  children,  
  
}) => (  
  <input  
    id={id}  
    value={value}  
    type={type}  
    onChange={onChange}/>  
  <div>  
    {children}  
  </div>  
)
```

```

    <>

    <label htmlFor={id}>{children}</label>

    &nbsp;
    <input
      id={id}
      type={type}
      value={value}
      onChange={onInputChange}
    />
  </>
);

```

src/App.js

Now the React component's elements behave similar to native HTML. Everything that's passed between a component's elements can be accessed as children in the component and be rendered somewhere. Sometimes when using a React component, you want to have more freedom from the outside what to render in the inside of a component:

```

const App = () => {
  ...

  return (
    <div>
      <h1>My Hacker Stories</h1>

      <InputWithLabel
        id="search"
        value={searchTerm}
        onChange={handleSearch}
      >

      <strong>Search:</strong>

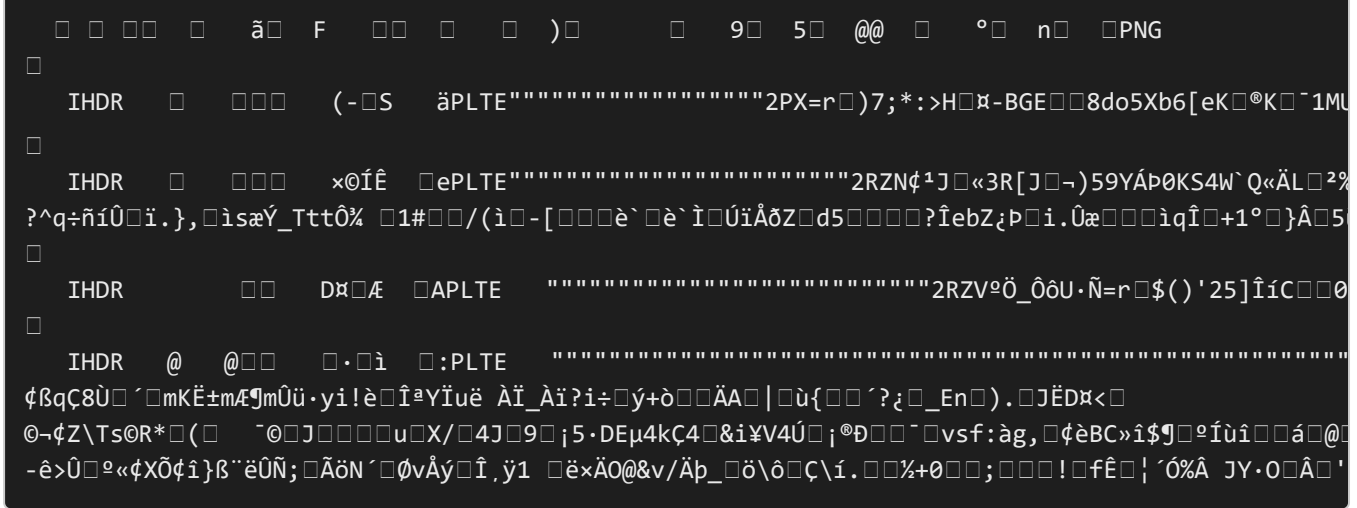
      </InputWithLabel>

      ...
    </div>
  );
};

```

src/App.js

With this React feature, we can compose React components into each other. We've used it with a JavaScript string and with a string wrapped in an HTML `<strong>` element, but it doesn't end here. You can pass components via React children as well.



## Exercises:

- Confirm the [changes from the last section](#).
- Read more about React Component Composition ([0](#), [1](#)).
- Create a simple text component that renders a string and passes it as `children` to the `InputWithLabel` component.