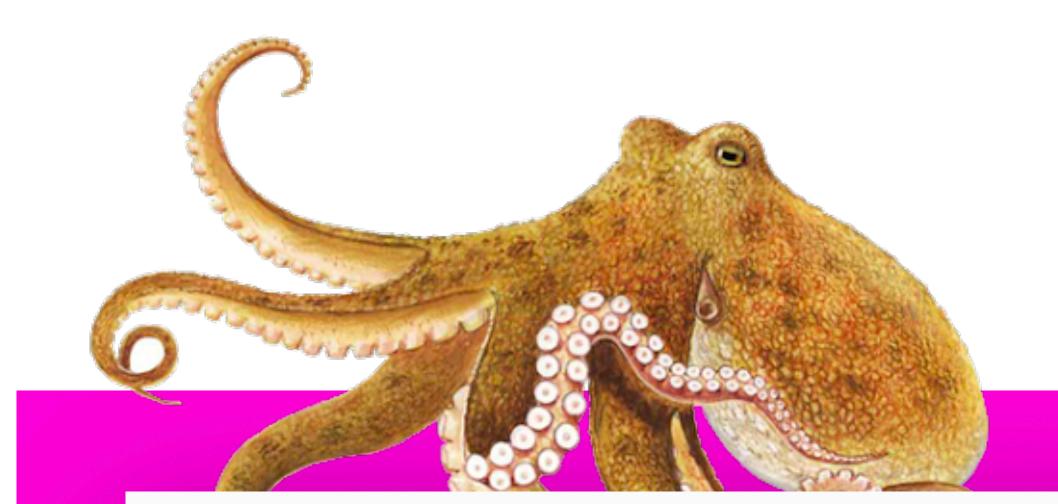
Render

Well render usage is pretty self explanatory, It renders the component.

Screen

Screen, gives us a set of utilities to Checkout if the component was rendered in the expected way



```
import { render, screen } from "@testing-library/react";
import App from "../App";

test('Render a component', ()=>{
   render(<App/>)
   const counter = screen.getByText(/counter/i);
   expect(counter).toHaveTextContent("Counter: 0");
})
```



• Automate user events

```
import { render, screen } from "@testing-library/react";
import userEvent from "@testing-library/user-event";
import App from "../App";
test('Render a component', ()=>{
  render(<App/>)
 const counter = screen.getByText(/counter/i);
 const increment = screen.getByRole("button", { name: /increment/i });
 const decrement = screen.getByRole("button", { name: /decrement/i });
 // We validate the initial render
 expect(counter).toHaveTextContent("Counter: 0");
 // We fire the click 🤌 🗑
 userEvent.click(increment);
 // Finally, we verify the results on the component
 expect(counter).toHaveTextContent("Counter: 1");
  // Same for decrement button
 userEvent.click(decrement);
  expect(counter).toHaveTextContent("Counter: 0");
```