

# Övning: Testning i React med Vitest

Enkel testning av komponenten `Greeting`.

## Installera dependencies

Öppna terminalen och kör:

```
npm install --save-dev vitest @testing-library/react @testing-library/jest-dom  
npm install --save-dev @types/jest  
npm install jsdom
```

Vi kör `vitest` som ramverk för testning med hjälp av biblioteken `jest` och `jsdom`

## Lägg till test-script i `package.json`

Öppna `package.json` och lägg till följande under `scripts` :

```
"scripts": {  
  //lägg till test-nyckel med värdet vitest  
  "test": "vitest"  
}
```

Det gör att du kan köra tester med:

```
npm test
```

## Skapa `vitest.config.ts` i projektets rotmapp

Skapa en konifgurationsfilen `vitest.config.ts` och lägg till följande:

```
import { defineConfig } from "vitest/config";

export default defineConfig({
  test: {
    globals: true,
    environment: "jsdom",
    setupFiles: "./src/setupTests.ts",
  },
});
```

## Steg 4: Skapa `setupTests.ts`

Skapa en fil `setupTests.ts` i `src/`.

Den här filen ska innehålla importers och annat som är gemensamma för våra test

```
import "@testing-library/jest-dom";
```

Filen laddas in automatiskt innan testerna körs.

## Steg 5: Skapa testmappen och testfilen

Skapa mappen `tests/` i `src/` och skapa testfilen `Greeting.test.tsx` med följande innehåll:

```
import { render, screen } from "@testing-library/react";
import { describe, test, expect } from "vitest";
import Greeting from "../components/Greeting";

describe("Greeting Component", () => {
  test("skriva ut Greeting med message prop", () => {
    render(<Greeting message="Hello world" />);
    const headingElement = screen.getByText("Hello world");
    expect(headingElement).toBeInTheDocument();
  });
});
```

## Vad händer i testfilen?

### Skapa en testgrupp med `describe` :

- `describe("Greeting Component", () => {...})` skapar en testgrupp för `Greeting` -komponenten.

### Skriva ett test med `test` :

- `test("skriva ut Greeting med message prop", () => {...})` beskriver vad vi testar.
- `render(<Greeting message="Håkan" />);` renderar komponenten med en prop `message="Håkan"` .
- `screen.getByText();` söker efter texten i DOM:en.
- `expect(headingElement).toBeInTheDocument();` kontrollerar att texten finns i DOM:en.

## Steg 6: Kör testerna

För att köra testerna, kör följande kommando:

```
npm test
```