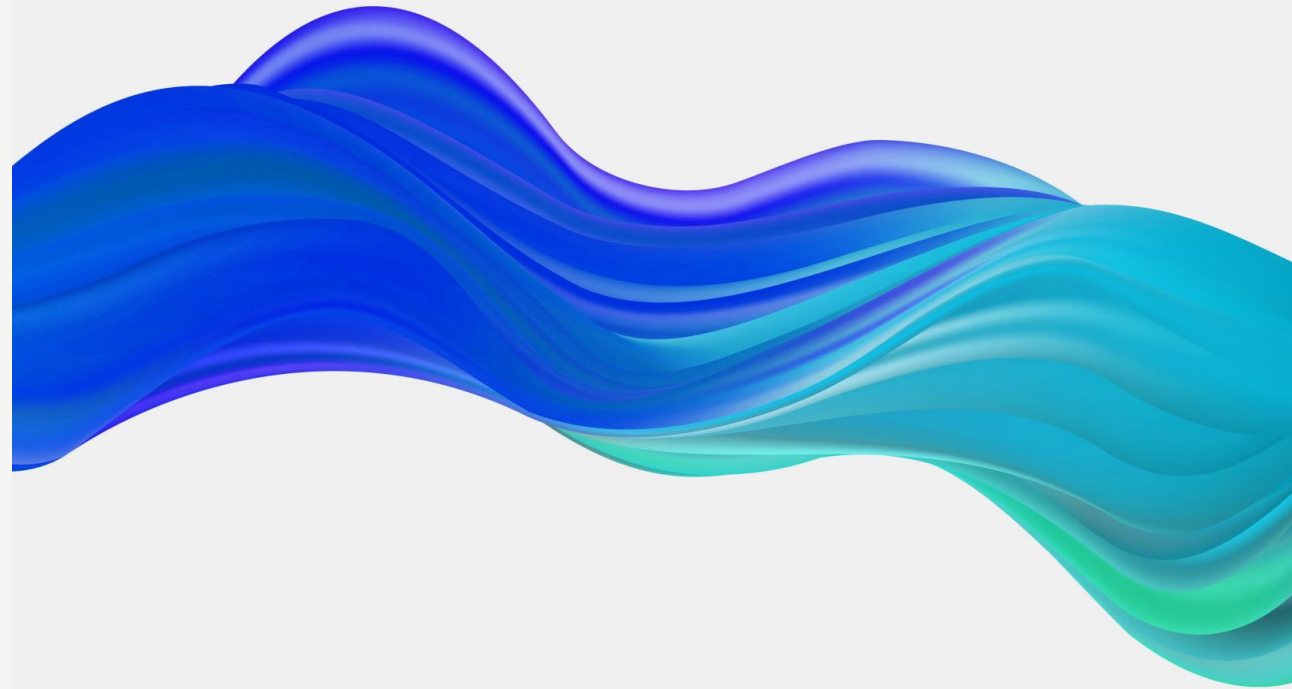


Teknisk testing: E2E i Cypress

End-to-end testing med Cypress
som verktøy



Kompetansemål vi jobber med i denne timen:

« anvende relevant testmiljø og utføre testing tilpasset IT-løsningen som utvikles »



Teknisk testing

Ulike typer «teknisk» testing:

- Unit testing
- Integrasjonstesting
- Snapshot-testing
- Functional testing
- **End-to-end testing**
- Acceptance testing
- Performace testing
- Smoke testing



E2E-testing

End-to-end testing

« End-to-end testing is a methodology used in the software development lifecycle (SDLC) to test the functionality and performance of an application under product-like circumstances and data to replicate live settings. The goal is to simulate what a real user scenario looks like from start to finish. »

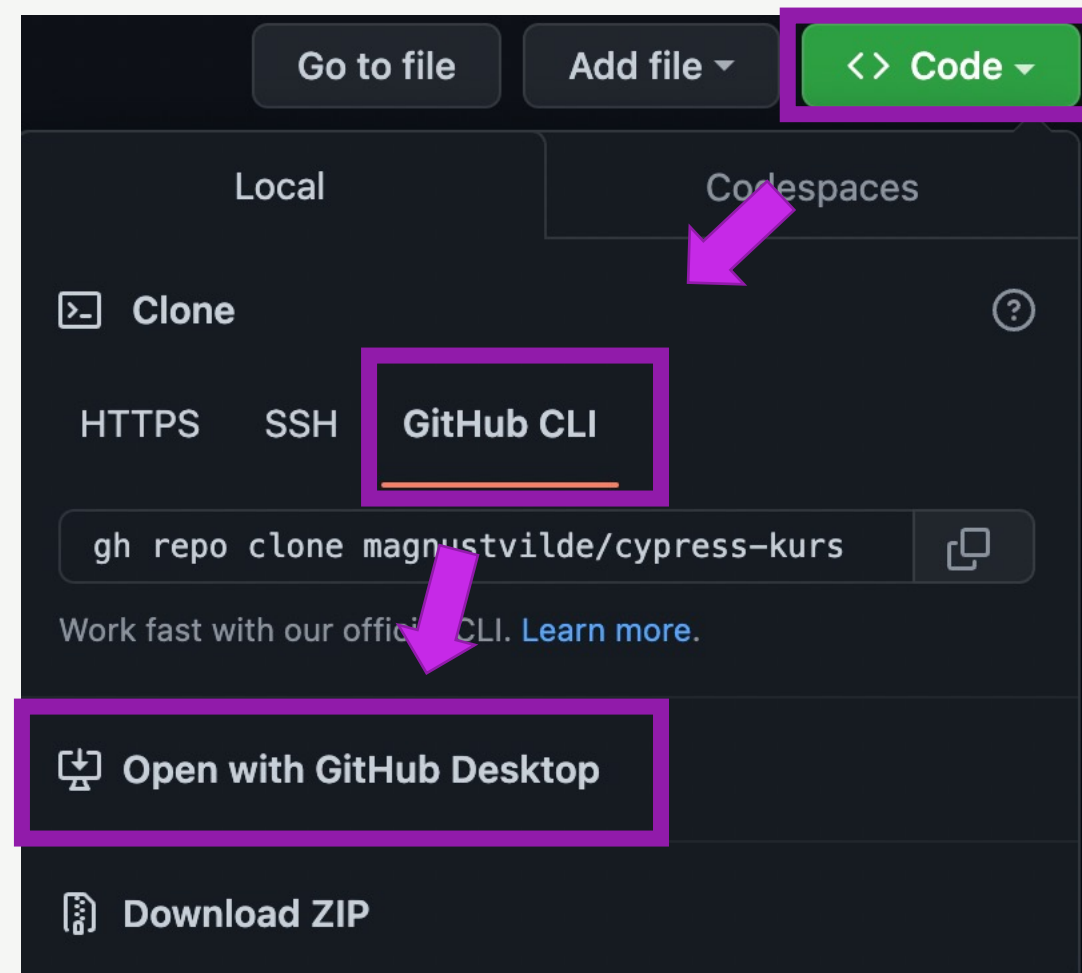


Hva er Cypress?

- <https://www.youtube.com/watch?v=BQqzfHQkREo>

Oppsett: Nettside å kjøre tester på

- Klone ned GitHub-repo til mappen din «Repositories»:
 - <https://github.com/magnustvilde/cypress-kurs>



Installere dependencies

- Naviger deg til mappen du har klonet
- **npm install** (kommandoen installerer alle node modules som trengs for prosjektet)



Oppsett: Installere Cypress

- **npm install cypress --save-dev** (legger inn alle pakker som trengs i Node-modules og legger til dependencies i din package.json-fil)



Hvordan kjøre:



- **npm run dev** (kjører server lokalt)
- **npx cypress open** (åpner en Cypress-kontrollert versjon av din nettleser der du kan bla deg mellom testene dine og se hvordan det gikk steg for steg.)

Ekstra alternativ, dersom du ønsker videoopptak og oppsummering av testene dine:

- **npx cypress run** (kjører Cypress i terminalen, og viser deg en oppsummering av hvordan testene gikk OG tar en video av kjøringen)
- (finnes også mange andre kommandoer, oversikt her:
<https://docs.cypress.io/guides/command-line>)

Hvordan lage E2E-tester: Dokumentasjon fra Cypress

- <https://docs.cypress.io/guides/end-to-end-testing/writing-your-first-end-to-end-test>



Hvordan lage E2E-tester: Vi gjør det sammen

cd vite-project

touch cypress/e2e/home_page.cy.js



Hvordan lage E2E-tester:

Vi gjør det sammen

```
describe('The Home Page', () => {  
  it('successfully loads', () => {  
    cy.visit('http://localhost:5173')  
  })  
})
```



Nyttige funksjoner i Cypress

Funksjoner for å sette opp tester:

it('beskrivelse' ,() => {})

- måten vi lager nye tester på

beforeEach(() => {})

- gjøres før HVERT it()-kall



Noen nyttige metoder i Cypress

Alle kommandoer starter med **cy**, gjøres inni `it()`

Metoder som legges etter **cy**



.contains('tekst')

- leter etter tekst i din DOM

.get('form')

- finner et element i HTML, kan også brukes med `.klasse` og `#id`

.get(parent child)

- finner et element child, som er inni et element parent

.click()

- trykker på et element fra din DOM

.type('tekst')

- skriver til et element, må 'chaines' til et element som kan skrives til

.on()

- fungerer som en «event-listener», f.eks. `'window:alert'`

Resten av timen/videre arbeid

- Finn et eget prosjekt du har holdt på med, og installer cypress i det prosjektet
- Sett opp fornuftige tester som sjekker om programmet oppfører seg som planlagt
- Let etter «edge-cases» – altså prøv å være en vanskelig og dum bruker som trykker på feil ting i testen din
- Lær deg hvordan «Component testing» fungerer