# Workshop II Copilot: Skriv Effektive Prompter til VSCode GitHub Copilot

## Mål

Ved afslutningen af denne workshop vil du:

* Forstå opbygningen af en effektiv Copilot-prompt
* Øve dig i at skrive prompts, der genererer præcis kode
* Lære at forfine prompts for bedre resultater

## Forudsætninger

* GitHub Copilot installeret og aktiveret i VS Code
* Grundlæggende programmeringsforståelse i JavaScript

## Prelab

**Grundlæggende Funktioner:bruge Copilot**

• Skriv en funktion ved navnprint, der tager en parameter name og logger en hilsen til konsollen.

• Skriv en funktion ved add\_num, der tager to number som parametre og returnerer deres sum.

//simple JavaScript function named greet that takes a parameter name and logs a greeting message to the console

**Spørgsmål:**

•Hvad er forskellen mellem parametre og argumenter i funktioner?

• Hvordan håndterer JavaScript funktioner med flere eller færre argumenter end parametre?

• Hvad er formålet med return-sætningen i en funktion?

## Øvelse 1: Prompt-Engineering (25 minutter)

Lad os se, hvordan forskellige detaljeringsniveauer i prompts påvirker Copilots output.

1. Opret en ny fil med html loginform.html (hente kode fra xbi) og ny javascript fil kaldet user\_validation.js
2. Prøv disse prompts i rækkefølge og observer, hvordan Copilots forslag forbedres:
   * **Basal Prompt:**

// validere email

* + **Bedre Prompt:**

// funktion til at validere email-format

* + **Bedste Prompt:**

// Funktion validateEmail der tjekker:

// - indeholder @ og mindst ét punktum

// - ingen mellemrum tilladt

// - minimumslængde på 5 tegn

**Diskussion**: Bemærk hvordan mere kontekst og krav giver mere præcise forslag.

## Øvelse 2: Forfining af Prompter (15 minutter)

Nogle gange giver din første prompt ikke det ønskede resultat. Lad os øve os i at forfine:

Bruge /new I github copilot: nye filen

user\_validation\_copy.js

1. Start med denne prompt:

// check adgangskode

2. Forfin den gradvist: checkPassword

// check sikker adgangskode

// check sikker adgangskode med:

// - minimum 12 tegn

// - mindst ét stort og ét lille bogstav

// - mindst ét tal og ét specialtegn

## Øvelse 3: Løs opgave fra Kode til Web I par(45 minutter)

For at styrke dine færdigheder i at bruge GitHub Copilot, skal du vælge en øvelse fra din lærebog og implementere den ved hjælp af Copilot i Visual Studio Code. Start med at skrive en kommentar, der beskriver opgaven, og lad Copilot foreslå koden. Evaluer og juster de genererede forslag efter behov. Afslut med at teste din løsning i liveserver.

<https://koder.systime.dk/?id=137>

## Øvelse 4: ekstra - JS Counter -

XBI JsCounter start version

New feature Jscounter saver

### ## Tips

- Vær specifik om funktionsnavne, parametre og returværdier

- Inkluder eksempler, hvis adfærden ikke er indlysende

- Brug klar formatering i dine prompts (punktopstillinger, linjeskift)

- Hvis du ikke kan lide Copilots forslag, prøv at start forfra / prøv andre AI

prompt//you will create a login form, with user name and password, then you will put validation in the password input field, if user has input 3 letters then output should be error, min passoword length 3 charaters, the errror messge should be efter the password field

## Refleksionsspørgsmål

1. Hvilket prompt-mønster fungerede bedst for dig?

2. Hvordan påvirkede detaljegraden i dine prompts Copilots output?

3. Hvilke forskelle bemærkede du mellem din oprindelige og din forfinede prompt?

---

Kilder:

Børne- og Undervisningsministeriet. (2022). *Didaktiske principper i erhvervsinformatik*. EMU. Retrieved January 29, 2025, from <https://emu.dk/eud/erhvervsinformatik/didaktiske-principper>

Reflection

\*\*Reflection Session: AI Prompt Learning in Programming\*\*

### \*\*Objective:\*\*

The purpose of this reflection session is to evaluate your experience using AI prompt learning for programming. By reflecting on your learning process, challenges, and outcomes, you can gain insights into how AI-assisted learning fits into your development as a programmer.

---

### \*\*Part 1: Personal Experience\*\*

1. \*\*Initial Thoughts:\*\* What were your expectations before using AI for programming assistance? Were you excited, skeptical, or unsure?

2. \*\*Ease of Use:\*\* How intuitive was it to craft effective prompts? Did you find it easy or challenging to get the AI to provide useful responses?

3. \*\*Learning Curve:\*\* Did AI assistance make it easier or harder to understand programming concepts? Why?

---

### \*\*Part 2: Effectiveness and Challenges\*\*

4. \*\*Problem-Solving:\*\* Did AI help you overcome coding challenges? If so, can you provide an example?

5. \*\*Accuracy of Responses:\*\* How accurate and helpful were the AI-generated responses? Did you notice any incorrect or misleading answers?

6. \*\*Critical Thinking:\*\* Did using AI encourage deeper learning, or did it make you more dependent on external help? Explain your reasoning.

7. \*\*Ethical Considerations:\*\* What are the ethical concerns of relying on AI for programming, such as plagiarism or lack of originality?

---

### \*\*Part 3: Application and Future Use\*\*

8. \*\*Integration in Learning:\*\* How do you see AI fitting into your future learning process? Would you continue using it, and if so, in what ways?

9. \*\*Balancing AI and Manual Coding:\*\* How can you balance AI assistance with developing your own problem-solving skills?

10. \*\*Feedback & Suggestions:\*\* What improvements could be made to AI prompt learning tools to enhance their usefulness in programming education?

---

### \*\*Conclusion:\*\*

Take a few moments to summarize your key takeaways from this experience. Has AI prompt learning changed the way you approach programming? Why or why not?

Use this reflection to refine your approach to AI-assisted programming and make the most of its benefits while addressing its challenges!