

Nieuw '25

- [Scratch 1](#)
- [Scratch 2](#)
- [Introductie AI](#)
- [Prompt Engineering 1](#)
- [Prompt Engineering 2](#)
- [Prompt Engineering 3](#)

Scratch 1

Introductie en start

Welkom! We gaan je stap voor stap helpen om Scratch te begrijpen. Zo leer je zelf programmeren. Maar wat gaan we eigenlijk doen?

Wat kun je verwachten?

- Je bekijkt korte uitlegvideo's van Feliene, een docent aan de TU Delft.
- Je maakt opdrachten direct in Scratch.
- Je leert stap voor stap hoe je een eigen doolhofspel bouwt.

Voorbereiding: Open dit project in Scratch:

<https://scratch.mit.edu/projects/96709199/#editor>

(of download het bestand hier: [Doolhof Start.sb3](#))

Je zou dit startscherm moeten zien:

image.png
image not found or type unknown

Bekijk de uitlegvideo

In deze video legt Feliene uit wat je gaat bouwen en hoe je aan de slag kunt. Let goed op de uitleg over hoe Scratch werkt.

<https://www.youtube.com/embed/JcIVNv4VZv8?si=8FT4Hu648QqKqrCY>

Alternatieve link: [Klik hier als de video hierboven niet werkt](#)

Opdracht

Beschrijf in één zin in je eigen woorden wat je denkt dat je gaat maken.

Inleveren

Typ je zin in het tekstvak hieronder. Gebruik je eigen woorden. Denk aan wat je in het spel gaat bouwen.

Stap 1 – Beweging

We beginnen met de besturing van je hoofdpersonage, Giga. In deze stap leer je hoe je Giga kunt laten bewegen met de pijltjestoetsen.

Wat leer je?

- Bewegen over de X-as (links en rechts).
- Bewegen over de Y-as (omhoog en omlaag).

Bekijk de uitlegvideo

<https://www.youtube.com/embed/IGtsy24vbiE?si=ZYnQwgoNBWrrlWw>

Alternatieve link: [Bekijk de video via HiDrive](#)

Opdracht

Hoeveel richtingen kan jouw poppetje bewegen aan het einde van deze stap?

Inleveren

Vul in het tekstvak in hoeveel richtingen jouw sprite beweegt.

Stap 2 – De lus (loop)

In deze stap leren we hoe je een herhaling maakt met een lus. Hiermee kun je bijvoorbeeld code telkens opnieuw uitvoeren zolang een bepaalde voorwaarde klopt.

Wat leer je? Je gebruikt een herhaalblok om acties te blijven herhalen.

Bekijk de uitlegvideo

<https://www.youtube.com/embed/OFzSUt23Lho?si=r63G16qrEFOA2YY>

Alternatieve link: [Bekijk de video via HiDrive](#)

Opdracht

Leg in je eigen woorden uit wat een lus (loop) doet in een programma.

Inleveren

Typ je uitleg over de lus in het tekstvak.

Stap 3 – Als-dan-anders

We voegen nu logica toe: wat moet er gebeuren als iets gebeurt? Je leert nu het *als-dan-anders* blok gebruiken.

Bekijk de uitlegvideo

<https://www.youtube.com/embed/6f9TeuFZshM?si=XWdXdtFBhWh0XuD8>

Alternatieve link: [Bekijk de video via HiDrive](#)

Opdracht

Wat doet een als-dan-anders (if-then-else) blok in je code? Leg het uit in je eigen woorden.

Inleveren

Typ je uitleg in het tekstvak.

Stap 4 – Terug naar startpositie

Als Giga een muur raakt, willen we dat hij teruggaat naar het begin. In deze stap leer je hoe je zijn positie reset.

Bekijk de uitlegvideo

https://www.youtube.com/embed/IRX_n2S0Hwk?si=YNeZtufi6l11jnJX

Alternatieve link: [Bekijk de video via HiDrive](#)

Opdracht

Welke coördinaten (X en Y) gebruikt jouw spel om Giga terug te zetten naar de start?

Inleveren

Typ de X- en Y-positie in het tekstvak.

Stap 5 – De sleutel

Je spel is bijna klaar! In deze stap voeg je een sleutel toe. Als Giga deze aanraakt, is het spel gewonnen.

Bekijk de uitlegvideo

<https://www.youtube.com/embed/UoqZhd4ggn0?si=j8a5GLqdgiNfATHc>

Alternatieve link: [Bekijk de video via HiDrive](#)

Opdracht

Maak een screenshot van jouw hele browser waarin je de gemaakte code laat zien.

Inleveren

Upload de screenshot met jouw Scratch-code.

Stap 6 – Monster toevoegen

We maken het spannend! Voeg een monster toe dat je moet ontwijken. Laat het monster bewegen.

Bekijk de uitlegvideo

<https://www.youtube.com/embed/6sYbPEse-jc?si=ugD4jr0WJ0vxC-8N>

Alternatieve link: [Bekijk de video via HiDrive](#)

Opdracht

Laat het monster bewegen. Lever een screenshot in van de code die het monster laat bewegen.

Inleveren

Upload de screenshot van de code van het monster.

Stap 7 – Game over

Als het monster Giga raakt, is het spel afgelopen. Je leert nu hoe je het spel opnieuw kunt laten starten.

Bekijk de uitlegvideo

<https://www.youtube.com/embed/LR5vDCgcLMM?si=Kj7SCUr9BADcQYGp>

Alternatieve link: [Bekijk de video via HiDrive](#)

Opdracht

Laat zien dat Giga teruggezet wordt naar het begin wanneer hij de muur of het monster raakt.

Inleveren

Screenshot van de code die dit laat zien.

Stap 8 – Alles terugzetten

Als het spel opnieuw begint, moeten ook het monster en Giga teruggezet worden naar hun beginpositie. Pas ook de snelheid aan als het spel te moeilijk wordt.

Bekijk de uitlegvideo

<https://www.youtube.com/embed/N76K2ZfCP3o?si=Yf6DBrKI71vHcx2V>

Alternatieve link: [Bekijk de video via HiDrive](#)

Opdracht

Pas de snelheid van het monster aan en laat zien hoe je alles terugzet.

Inleveren

Screenshot van de code waarin het monster wordt teruggezet én je de snelheid hebt aangepast.

Stap 9 – Punten bijhouden

We voegen nu een score toe met behulp van een variabele.

Een variabele is een waarde die je kunt opslaan en aanpassen tijdens het spel.

Bekijk de uitlegvideo

https://www.youtube.com/embed/1GTrDEd1ECY?si=_ALBcbZ-dacTo0dq

Alternatieve link: [Bekijk de video via HiDrive](#)

Opdracht

Laat zien hoe je de score bijhoudt.

Inleveren

Screenshot van je code waarin je score bijhoudt met een variabele.

Stap 10 – Afronden

We maken het spel af en zorgen dat het er goed uit ziet. Je kunt je spel opslaan op je laptop of online delen.

Bekijk de uitlegvideo

<https://www.youtube.com/embed/cHWgQNulLjU?si=5iQpiZdUwrxSB8UU>

Alternatieve link: [Bekijk de video via HiDrive](#)

Opdracht

Maak het spel af, voeg iets unieks toe en sla het op.

Inleveren

Lever jouw eigen unieke versie van het spel in.

Stap 11 – Wat heb je geleerd?

Je hebt veel geleerd over programmeren. Nu kijk je terug op wat je allemaal hebt gedaan.

Opdracht

- **Vraag 1:** Waarvoor heb je in dit spel een lus gebruikt?
- **Vraag 2:** Waarvoor heb je een als-dan-anders blok gebruikt?
- **Vraag 3:** Waarvoor heb je een variabele gebruikt?

Inleveren

Schrijf je drie antwoorden op en lever ze in.

Scratch 2

Links

1. <https://www.youtube.com/watch?v=zRWO9vWytPA>
2. <https://www.youtube.com/watch?v=iw7YvogC5Uo>
3. https://www.youtube.com/watch?v=vOAI_YY6Ntk
4. https://www.youtube.com/watch?v=L8q4OR8n_yw
5. https://www.youtube.com/watch?v=mluq9_b4xSI
6. <https://www.youtube.com/watch?v=Cg9T9Hol7tY>
7. <https://www.youtube.com/watch?v=UDpxvbCCMHA>
8. <https://www.youtube.com/watch?v=CQby8IzbkZg>
9. <https://www.youtube.com/watch?v=dM9a4XprB7o>
10. <https://www.youtube.com/watch?v=CHWy9ILV4bA>
11. <https://www.youtube.com/watch?v=dV4ivXfP2q0>

Introductie AI

Introductie

In deze les leren we wat AI is en we gaan kijken naar het verschil van programmeren met en zonder AI.

We kijken naar de kracht van AI maar ook naar de tekortkomingen.

We gaan de volgende dingen leren:

1. Wat is AI?
2. Wat zijn de AI toepassingen?
3. AI versus klasieke ('gewone') code. Wat is het verschil?
4. Voordelen van AI
5. Nadelen van AI
6. Hoe kan je AI slim gebruiken
7. Prompt Engineering (intro)

1, wat is AI?

Bekijk deze video:

https://www.youtube.com/watch?v=QJE_ycgR8E8

https://www.youtube.com/embed/QJE_ycgR8E8

Opdracht 1

Vat in één tot drie zinnen samen wat de kernboodschap van dit filmpje is.

Inleveren

1. Maak de samenvatting in een text document en vul deze in. Lever een **TXT** document in.

2, AI toepassingen

AI is veel meer dan alleen ChatGPT. In deze video wordt uitgelegd waarvoor AI kan worden gebruikt.

Bekijk deze video:

<https://www.youtube.com/watch?v=stw2upLHCuI>

<https://www.youtube.com/embed/stw2upLHCuI>

Theorie

AI-toepassingen per taaktype

Hieronder staan concrete voorbeelden van hoe kunstmatige intelligentie (AI) wordt toegepast in verschillende soorten taken

♦ 1. Classificatie

- **Wat is het?:** Het toewijzen van gegevens aan een bepaalde categorie.
- **Voorbeeld:** Een e-mailsysteem dat automatisch bepaalt of een e-mail *spam* is of *geen spam*, op basis van de inhoud, afzender en gebruikte woorden.

♦ 2. Associatie

- **Wat is het?:** Het ontdekken van patronen of combinaties van items die vaak samen voorkomen.
- **Voorbeeld:** Een webshop gebruikt AI om te ontdekken dat klanten die een *laptop* kopen ook vaak een *laptopshoes* kopen. Op basis daarvan worden aanbevelingen gedaan: "Andere klanten kochten ook..."

♦ 3. Optimalisatie

- **Wat is het?:** Het vinden van de beste oplossing uit veel mogelijkheden, vaak onder bepaalde voorwaarden.
- **Voorbeeld:** Een AI-systeem voor routeplanning bepaalt de *snelste bezorgroutes* voor een pakketdienst, rekening houdend met afstand, verkeer en bezorgtijd.

✦ 4. Voorspelling

- **Wat is het?:** Het voorspellen van toekomstige waarden of gebeurtenissen op basis van eerdere gegevens.
- **Voorbeeld:** Een bakker gebruikt AI om op basis van eerdere verkoopdata te voorspellen hoeveel *brood* er de komende week nodig is.

✦ 5. Creatie

- **Wat is het?:** Het genereren van nieuwe inhoud of ideeën met behulp van AI.
- **Voorbeeld:** Een AI-systeem zoals ChatGPT of DALL·E kan een *gedicht schrijven* of een *afbeelding maken* op basis van een beschrijving, bijvoorbeeld: "Een robot die schildert in een zonnebloemenveld".

Opdracht

Bepaal van elk van de voorbeelden bij welk type AI-toep[assing (creatie, associatie, optimalisatie, voorspellen, creatie) dit hoort.

1. Netflix geeft je aanbevelingen voor films op basis van wat je eerder hebt gekeken.
2. Een game bepaalt of je gedrag verdacht is en je mogelijk aan het valsspelen bent.
3. Een routeplanner voor je fietsrit kiest de route met de minste verkeerslichten.
4. TikTok voorspelt welke video je het langst gaat kijken en laat die eerder zien.
5. Een AI-programma maakt een unieke profielfoto in cartoonstijl van jou.

Inleveren

1. Neem de punten over in een text document en vul deze in. Lever een **TEXT** document in.

3, AI en 'gewone' computer code.

Wat is het verschil tussen 'gewone' code en AI-code?

Gewone code (klassieke algoritmes) is gebaseerd op vaste instructies: als je A invoert, gebeurt altijd B. Dit maakt het **voorspelbaar** en **betrouwbaar**. Denk aan een rekenmachine of een robotarm in een fabriek die elke minuut exact dezelfde beweging maakt. De computer voert precies uit wat je hebt geprogrammeerd.

AI-code werkt anders. Die is **getraind op heel veel voorbeelden** (zoals tekst, beelden of data) en leert daarvan zelf **patronen te herkennen**. Dat lijkt een beetje op hoe onze hersenen leren. AI is vaak **minder voorspelbaar**, omdat het zelf beslissingen neemt op basis van wat het geleerd heeft. Hierdoor kan het ook **fouten maken**, zeker als het iets nog niet eerder gezien heeft. ChatGPT is bijvoorbeeld een AI die voorspelt welk woord het beste past, op basis van miljarden voorbeelden.

Opdracht

Opdracht: AI-code of klassieke code? Lees de eigenschappen hieronder. Bepaal of het hoort bij klassieke code of bij AI-code. Zet er een kruisje bij:

Eigenschap	Klassieke code	AI-code	Geen van beiden
Voert altijd precies dezelfde handeling uit			
Kan leren van voorbeelden			
Maakt soms fouten bij onbekende situaties			
Is goed in rekenen en logica			
Kan patronen herkennen			
Is 100% voorspelbaar			
Kan nieuwe dingen maken (zoals een tekening)			
Kan adviseren of je in Bitcoin moet stappen of moet verkopen			
Zal altijd goed advies geven voor de aankoop/verkoop van Bitcoin			
Kan jouw foto veranderen en jou in een hele vreemde situatie zetten			

Inleveren

1. Neem de tabel over in een Word document en vul deze in. Lever een **PDF** document in.

4, voordelen van AI

Voordelen van AI

AI heeft een aantal sterke kanten die het nuttig maken in allerlei toepassingen:

- **AI kan grote hoeveelheden data analyseren** en daarin snel patronen ontdekken die mensen zouden missen.
- **AI werkt 24/7**, zonder pauzes of vermoeidheid.
- **AI kan gepersonaliseerde aanbevelingen geven**, bijvoorbeeld op YouTube of Spotify.
- **AI helpt bij medische diagnoses**, doordat het patronen in röntgenfoto's of scans herkent.
- **AI automatiseert saaie of gevaarlijke taken**, zoals kwaliteitscontrole in fabrieken of het inspecteren van pijpleidingen met drones.

Opdracht

Hieronder zie je een aantal situaties. Geef per situatie aan of AI hier een voordeel zou kunnen bieden, en leg uit waarom.

1. Een docent moet elke week 200 toetsresultaten controleren op fouten.
2. Een leerling zoekt elke dag naar nieuwe muziek die past bij zijn smaak.
3. Een ziekenhuis wil sneller afwijkingen op longfoto's opsporen.
4. Een bedrijf wil weten welke producten waarschijnlijk snel uitverkocht raken.
5. Een bakker wil weten hoeveel broden hij volgende week moet bakken.

Inleveren

1. Neem de punten over in een text document en vul deze in. Lever een **TXT** document in.

5, nadelen van AI

Nadelen van AI

Hoewel AI veel voordelen heeft, zijn er ook belangrijke nadelen en aandachtspunten:

- **AI kan fouten maken** als het situaties tegenkomt die het niet kent of niet goed begrijpt.
- **AI is afhankelijk van data** — als de data onvolledig of bevooroordeeld is, kan het systeem verkeerde beslissingen nemen.
- **AI is vaak een 'black box'** — het is soms moeilijk te begrijpen waarom een AI iets doet of beslist.
- **AI kan banen vervangen**, vooral bij repetitieve taken, wat zorgt voor zorgen over werkgelegenheid.
- **AI heeft geen ethiek of gevoel** — het kan geen morele afwegingen maken zoals mensen dat doen.
- **AI kan misbruikt worden** — bijvoorbeeld voor deepfakes, spam of het beïnvloeden van meningen via sociale media.

Opdracht: Waar is AI een risico of nadeel?

Bekijk de onderstaande situaties. Geef per situatie aan of je denkt dat AI hier een **risico of nadeel** kan zijn, en leg kort uit waarom.

1. Een AI-systeem beoordeelt sollicitaties automatisch en kiest wie wordt uitgenodigd.
2. Een AI-chatbot geeft medisch advies zonder dat een arts meekijkt.
3. Een zelfrijdende auto moet een noodbeslissing nemen in het verkeer.
4. Een bedrijf gebruikt AI om te controleren hoeveel pauze werknemers nemen.
5. Een leerling gebruikt AI om al zijn schoolopdrachten te laten schrijven.

Inleveren

1. Neem de punten over in een text document en vul deze in. Lever een **TEXT** document in.

6, slim gebruik van AI

Hoe benut je de voordelen en vermijd je de nadelen?

AI is een krachtig hulpmiddel, maar het is belangrijk dat je er **bewust en slim mee omgaat**. Dat betekent: weten wanneer je AI goed kunt inzetten, en ook herkennen wanneer het beter is om zelf na te denken of iets te controleren.

✓ Zo benut je de voordelen van AI:

- Gebruik AI als **assistent**, niet als vervanger van je eigen denkwerk.
- Laat AI je **helpen bij brainstormen**, schrijven of samenvatten — maar **controleer altijd de output**.
- Gebruik AI om **saaie of repetitieve taken te versnellen**, zoals opmaak of vertalingen.
- Combineer AI-output met je **eigen kennis en creativiteit**, zodat het persoonlijk blijft.

✗ Zo vermijd je de nadelen van AI:

- **Geloof niet alles wat AI zegt** — controleer feiten en cijfers.
- **Gebruik AI niet voor belangrijke beslissingen** zonder menselijk toezicht.
- **Denk na over privacy en veiligheid** — deel geen persoonlijke gegevens.
- **Gebruik AI niet om te spieken** of werk van anderen als je eigen werk in te leveren
- **Weet wanneer je AI niet moet gebruiken**: Soms is menselijk oordeel belangrijker, bijvoorbeeld bij gevoelige onderwerpen of ethische keuzes.

Opdracht: Slim of niet slim gebruik van AI?

Lees de onderstaande situaties. Geef per situatie aan of dit **slim gebruik van AI** is of juist **onverstandig**, en leg uit waarom.

1. Een leerling vraagt ChatGPT om 5 ideeën voor een spreekbeurt en kiest daarna zelf het beste idee.

2. Iemand plakt een volledige schoolopdracht in ChatGPT en levert het antwoord in zonder iets aan te passen.
3. Een student laat AI een samenvatting maken van een moeilijke tekst en controleert die daarna met de originele tekst erbij.
4. Iemand vraagt aan een AI wat de beste medicijnen zijn voor zijn klachten, zonder met een arts te praten.
5. Een leerling gebruikt AI om zijn code te verbeteren, maar probeert eerst zelf te begrijpen wat het doet.

Inleveren

1. Neem de punten over in een text document en vul deze in. Lever een **TXT** document in.

7, *Prompt engineering*

Prompt engineering is het slim en bewust formuleren van opdrachten of vragen (prompts) voor een AI-systeem zoals ChatGPT, zodat je een zo goed mogelijk en bruikbaar antwoord krijgt.

Als jij aan een mede student vraagt of hij je met de vorige opdracht kan helpen, dan weet hij waarschijnlijk waar je het voer hebt. Een AI zoals ChatGPT weet dat niet. Omdat ChatGPT de context niet weet.

De context is alles om een vraag heen.

Voorbeeld

“Maak een quiz.”

Deze opdracht is vaag. De AI weet niet:

- Over welk onderwerp?
- Voor welke doelgroep?
- Hoeveel vragen?
- Meerkeuze of open vragen?

Voorbeeld met goede prompt engineering:

“Maak een quiz van 5 meerkeuzevragen voor leerlingen van 14 jaar over het onderwerp 'kunstmatige intelligentie'. Geef bij elke vraag vier antwoordopties en geef aan welk antwoord het juiste is.”

Deze prompt:

- geeft **context** (leeftijd, onderwerp)
- is **duidelijk** (5 vragen, meerkeuze)
- is **doelgericht** (quiz maken)
- bevat extra **details** (aantal opties en juiste antwoord)

We gaan hier in de module prompt engineering verder mee werken, maar we gaan nu vast een oefening doen.

Opdracht, maak een prompt

Maak één prompt waarin je je AI vraagt een PHP programma dat zoveel mogelijk lijkt op:

image.png
Image not found or type unknown

Je hoeft geen database te maken, je hoeft alleen front-end code te maken.

Maak dit met één prompt en wees zo compleet mogelijk.

Inleveren

1. De door jouw gemaakte prompt
2. Het resultaat in code

Prompt Engineering 1

Introductie

Prompt Engineering is alsof je een buitenstaander die niets van je weet iets wilt vragen. Omdat te doen moet je dus duidelijk en goed communiceren.

image.png
image not found or type unknown

We gaan leren hoe we goede prompt kunnen schrijven.

Voor een **goede prompt** moet je rekening houden met de volgende zaken:

1. **Context** - een goede prompt heeft voldoende context.
2. **Details/Specifiek** - een goede prompt heeft voldoende details en is zo specifiek mogelijk.
3. **Duidelijkheid** - een goede prompt is duidelijk.
4. **Doelgericht** - een goede prompt is doelgericht.
5. **Vorm** - in een goede prompt kan je de output in een bepaalde vorm vragen.
6. **Toon** - door in de prompt de toon op te nemen, bepaal je de vorm van het antwoord.

Al deze punten worden stuk-voor-stuk behandeld. De eerste drie zijn **rood** omdat *élke* prompt altijd moet voldoen aan deze kenmerken.

Maar laten we eerst even kijken wat nu eigenlijk *prompting* precies is.

Wat is prompting?

Prompting betekent dat je iets vraagt aan een AI, zoals ChatGPT, op een manier waardoor je een goed antwoord krijgt. Je *geeft een opdracht of stelt een vraag*, en dat noemen we een **prompt**.

Je kunt het vergelijken met hoe je een klasgenoot iets vraagt:

1. Als je gewoon zegt: "Doe dit," snapt die ander misschien niet precies wat je bedoelt.

2. Maar als je zegt: "Kun je me helpen met deze code, want ik snap het niet?", is dat al een stukje beter.
3. Maar het kan nog beter: "Kun je me helpen met deze code want ik krijg een foutmelding 'unknown variable on line 23', ik snap niet wat de foutmelding betekent? "".
4. En zelfs dit kan nog beter want je vergeet een belangrijk detail, zie jij welke detail?

Kort gezegd: prompting is het slim stellen van een vraag of opdracht aan AI, zodat je krijgt wat je nodig hebt.

In de laatste zin staat niets over in welke omgeving en met welke programmeertaal je werkt. Ook zou je kunnen overwegen om de code of een stuk daarvan mee te sturen.

Opdracht 1

Verbeter de prompt die hierboven bij punt 4 staat.

1. Context

We hebben geleerd dat een goede prompt detail en context heeft.

Context is een soort van omgeving.

Als je vraagt om code aan passen dan moet je dus aangeven wat de 'omgeving' is, in dit geval PHP en als het je bepaalde frameworks gebruikt of andere zaken die van belang zijn dan noem je die ook.

- **Context** betekent: de omgeving waarin iets zich afspeelt.
- Bij programmeren is dat bijvoorbeeld: de taal (zoals PHP), het framework (zoals Laravel), of iets anders dat belangrijk is om te weten.
- Een AI snapt je vraag beter als jij eerst vertelt wat de **omgeving/context** is.

Opdracht 2

Situatie

Je hebt de volgende PHP-code gekregen.

Deze code toont de huidige datum:

```
<?php  
echo date("Y-m-d");  
?>
```

Maar jij wil dat de datum verschijnt in het **Nederlands**, zoals: 9 mei 2025

De AI moet je helpen om de code aan te passen.

Jouw taak:

1. **Schrijf een prompt aan ChatGPT** waarin je vraagt om deze code aan te passen zodat de datum in het Nederlands verschijnt.
2. Let erop dat je **duidelijk aangeeft wat de context is**:
 - Je werkt met **PHP**
 - De code moet draaien op een gewone server (geen framework zoals Laravel)
 - Je wil de **datum in het Nederlands**

Inleveren

1. Screenshot van de prompt en het antwoord dat je hebt gekregen.
2. Screenshot van het resultaat in je browser.

2. Detail/Specifiek

Doel

Je leert hoe belangrijk **detail en specificiteit** zijn in een goede prompt. Hoe specifieker je bent, hoe beter de AI je kan helpen met het schrijven of verbeteren van PHP-code.

Theorie

- Een vage prompt zoals “*Maak een contactformulier*” geeft een vaag resultaat.

- Welke velden moeten erin?
 - Wat moet er met de ingevulde gegevens gebeuren?
 - Moet het formulier controle uitvoeren?
 - Moet er een succesbericht komen?
- Een **specifieke prompt** vertelt precies wat je wil:

Opdracht 3

Situatie

Je wil een **contactformulier in PHP** laten maken door ChatGPT.

Jouw taak

1. **Schrijf een prompt aan ChatGPT** waarin je duidelijk vraagt om een contactformulier in PHP.

Je moet in je prompt minimaal deze 4 dingen **specifiek vermelden**:

- Welke velden het formulier moet hebben (bv. naam, e-mail, bericht)
- Wat er met de data moet gebeuren (bijv. opslaan, versturen, tonen)
- Of er foutcontrole moet zijn (bijv. verplicht invullen, geldig e-mailadres)
- Of er een succesmelding moet komen na het verzenden

Inleveren

1. Screenshot van de prompt en het antwoord dat je hebt gekregen.
2. Screenshot van het resultaat in je browser.

3. Duidelijkheid

Duidelijk betekent dat je prompt voldoende detail bevat. Hierdoor is het duidelijk wat je precies bedoeld.

Een **duidelijke prompt** zegt precies:

1. *Wat* je wil
2. *Waarvoor* je het wil
3. Wat de beperkingen zijn
4. En eventueel *hoe* je het eruit wil laten zien

Een **voorbeeld** van de punten zijn:

1. Je wilt een website
2. Je wilt een website waarin de gebruiker zijn naam en adres moet opgeven.
3. Alle velden moeten op één web pagina passen en bestaat uit HTML en CSS.
De velden zijn allemaal verplicht. Zodra de submit button wordt ingedrukt wordt gecontroleerd of alle velden zijn ingevuld. Als dat niet het geval is dan volgt er een aanwijzing/waarschuwing.
4. Het moet er eenvoudig en professioneel uitzien, Gebruik 'Tailwind' als CSS framework.

Opdracht 4

Situatie

Je wil een AI vragen om een klein PHP-script te maken waarmee je een getal controleert: is het **even of oneven**?

Hieronder zie je een voorbeeld van een **vage prompt**. Lees hem goed.

“Schrijf iets in PHP dat met getallen werkt.”

Je gaat nu zelf een **duidelijke prompt** schrijven waarin je vraagt om een PHP-script dat:

- Één getal controleert
- Zegt of het getal even of oneven is
- De gebruiker het getal zelf laat invoeren via een HTML-formulier

Inleveren

1. Screenshot van de prompt en het antwoord dat je hebt gekregen.

2. Screenshot van het resultaat in je browser.

4. Doelgericht

Theorie

- AI is niet helderziend. Als je alleen zegt “Leg dit uit”, weet de AI niet hoe uitgebreid, voor wie of in welke vorm.
 - Wat wil je *precies* weten of krijgen?
 - Wil je code, uitleg, een stappenplan, een voorbeeld, een tabel, enz.?
 - Voor wie is het bedoeld? (Bijv. jezelf, een beginner, een klasgenoot?)Een **doelgerichte prompt** maakt duidelijk:

Opdracht

Situatie

Je wil begrijpen hoe een **while-loop** werkt in PHP, en je wil dat de AI het aan je uitlegt.

Deel 1: Vergelijk twee prompts

Prompt A (niet doelgericht):

“Wat is een while-loop?”

Prompt B (wel doelgericht):

“Leg uit wat een while-loop in PHP doet. Geef een voorbeeld met code en uitleg in simpele taal, zodat ik het kan gebruiken in een quiz voor klasgenoten.”

 Wat is het verschil tussen A en B?

Deel 2: Schrijf je eigen doelgerichte prompt

Bedenk nu een onderwerp in PHP dat jij lastig vindt (bijvoorbeeld: arrays, forms, functies, POST vs GET...).

Schrijf een doelgerichte prompt waarbij je duidelijk maakt **wat je wil, in welke vorm, en voor wie het bedoeld is**

Inleveren

1. Deel 1, leg in je eigen woorden uit wat het verschil is tussen prompt A en prompt B.
2. Deel 2, schrijf je eigen doelgerichte prompt (zie beschrijving deel 2).

5. Vorm

□ Theorie (kort samengevat)

- Soms weet de AI wél wat je bedoelt, maar krijg je het antwoord in een **onhandige vorm**:
 - Alleen tekst terwijl jij voorbeeldcode wilde.
 - Een lang verhaal terwijl jij liever een stappenplan had.
 - Code zonder uitleg erbij.
- Daarom is het slim om in je prompt te zeggen **hoe je het antwoord wil zien**:
 - Als lijst, tabel, voorbeeldcode, uitleg in korte zinnen, enz.

Opdracht

Deel 1: Slechte prompt (voorbeeld)

“Leg uit hoe je een formulier maakt met PHP.”

Wat is hier onduidelijk over de **vorm** van het antwoord?

Deel 2: Schrijf zelf een betere prompt

Jij gaat nu een betere prompt schrijven waarin je **duidelijk zegt in welke vorm** je het antwoord wil krijgen. Kies een vorm, zoals:

- Stap-voor-stap uitleg
- PHP-code met uitleg onder elke regel
- Een tabel met onderdelen van een formulier
- Een combinatie van uitleg en voorbeeld

☐ Vergeet niet te vermelden dat je werkt met **PHP**, en dat het voor een **beginner** is.

Inleveren

....

6. Toon

☐ Theorie

- De **toon** is *hoe iets klinkt* of *overkomt*: serieus, grappig, formeel, simpel, kinderlijk, informeel, enz.
- Als je geen toon aangeeft, kiest de AI zelf. Soms is dat te zakelijk, te moeilijk of juist te speels.
- In een goede prompt zeg je dus **hoe de AI zich moet gedragen**.

☐ Opdracht

Situatie:

Je wil dat ChatGPT uitlegt wat een **functie in PHP** is.

Je gaat dezelfde uitleg in **verschillende tonen** laten geven.

Stap 1: Schrijf drie verschillende prompts

Gebruik hetzelfde onderwerp ("Wat is een functie in PHP?") en verander **alleen de toon**.

1. ☐ **Formeel en technisch** (voor iemand met programmeerervaring):

2. **Grappig en speels** (alsof je het uitlegt aan een kind van 12):
3. **Duidelijk en vriendelijk** (voor een klasgenoot die net begint met PHP):

Inleveren

1. Welke toon vond jij het **meest duidelijk** voor een beginner?
2. Welke toon past het best bij jouw klasgroep?
3. Welke toon past het best bij jouw klasgroep?
4. Wat gebeurt er als je géén toon vraagt in je prompt?

Samengevat

Een goede prompt.

1. **Context** – Geef achtergrondinformatie, zoals *voor wie het is, wat het doel is, of waar het over moet gaan (welke programmeertaal?)*.
2. **Detail** – Hoe specifieker je bent, hoe beter het antwoord past bij wat je zoekt.
3. **Duidelijkheid** – Gebruik duidelijke en begrijpelijke taal. Vermijd vage of dubbelzinnige zinnen.
4. **Doelgerichtheid** – Geef aan *wat je precies wil* (bijvoorbeeld: een uitleg, een lijst, een verhaaltje, een vergelijking, enz.).
5. **Vorm** – Soms helpt het als je zegt *hoe* het antwoord eruit moet zien (bijvoorbeeld: “maak er een tabel van”, “gebruik korte zinnen”, “schrijf het op het niveau van een brugklasser”).
6. **Toon/Stem** – Wil je dat het grappig is? Serieus? Zakelijk? Kinderlijk? Dat kun je ook zeggen in je prompt.

Voorbeeld prompt

“Leg uit wat een if-statement is in PHP. Geef een simpel voorbeeld met uitleg in makkelijke taal. Ik ben 14 en net begonnen met PHP, dus graag zonder moeilijke woorden. Laat ook zien wat er gebeurt als de voorwaarde niet waar is.”

Opdracht 3

Maak een prompt waarin je aan AI vraagt om een programma in HTML-pagina te maken die er zo goed mogelijk lijkt op deze website template.

image.png
image not found or type unknown

Probeer dit in één tekst prompt te doen (gebruik dus géén plaatje) en denk aan alle 6 eigenschappen voor een goede prompt.

Inleveren

1. Prompt die je hebt gemaakt (schermafdruck).
2. Resultant (schermafdruck van de webpage).

Prompt Engineering 2

1, *Introductie*

We hebben in prompt engineering 1 geleerd waaraan een goede prompt moest voldoen.

Dit zijn de basis kenmerken van een goede prompt. De eerste drie kenmerken moet je prompt **altijd** aan voldoen!

1. **Context** - een goede prompt heeft voldoende context.
2. **Details/Specifiek** - een goede prompt heeft voldoende details en is zo specifiek mogelijk.
3. **Duidelijkheid** - een goede prompt is duidelijk.
4. **Doelgericht** - een goede prompt is doelgericht.
5. **Vorm** - in een goede prompt kan je de output in een bepaalde vorm vragen.
6. **Toon** - door in de prompt de toon op te nemen, bepaal je de vorm van het antwoord.

In deze module gaan we 13 **advanced prompt-technieken** leren. Deze technieken heb je niet altijd nodig maar het is handig om deze technieken te kennen.

Bovendien zijn de meeste technieken ook toepasbaar in als '**problem solving**' technieken.

□□ Prompting technieken 1

(laagdrempelig, direct toepasbaar)

1. **Isolate the problem**
Focus alleen op het onderdeel dat opgelost moet worden.
Laat overbodige context of code weg, zodat de AI zich op het juiste richt.
2. **Provide lists in bullet points**
Structuur helpt de AI om overzichtelijke en duidelijke antwoorden te geven.

3. **Provide the order if you ask for multiple tasks**
Geef een logische volgorde bij samengestelde opdrachten.
4. **Geef voorbeelden (few-shot prompting)**
Laat zien wat je bedoelt met een input/output-voorbeeld.

Intermediate prompting technieken

(voor betere controle en outputkwaliteit)

6. **Stel voorwaarden of beperkingen**
Geef grenzen aan zoals "gebruik max. 100 woorden" of "geen disclaimers".
7. **Werk in stappen (chain-of-thought prompting)**
Vraag de AI om stap voor stap te redeneren of eerst een plan te maken.
8. **Gebruik een rol (role prompting)**
Laat de AI een specifieke rol of expertise aannemen (bijv. leraar, programmeur, leerling).
9. **Vraag om alternatieven of variaties**
Stimuleer de AI om meerdere ideeën of invalshoeken te geven.

Geavanceerde prompting technieken

(voor reflectie en optimalisatie)

10. **Iteratief verbeteren**
Vraag de AI om haar eigen antwoord te verbeteren of herschrijven.
11. **Gebruik prompt templates**
Werk met vaste structuren zoals: context, doel, outputvorm, toon.
12. **Vraag om zelfvragen (self-questioning)**
Laat de AI zelf bedenken welke vragen nodig zijn om het probleem goed op te lossen.
13. **Gebruik kritiek als leermoment (prompt debugging)**
Laat de AI uitleggen waarom een prompt niet goed werkte en hoe je die kunt verbeteren.

1. Isolate the problem

Uitleg

Richt je prompt op het exacte probleem. In plaats van een hele codepagina te geven, geef alleen het stuk code of de situatie waar het om draait. Hoe minder ruis, hoe beter de AI kan helpen.

Je ziet dus dat je voor een goede prompt de code goed moet kunnen lezen.

Opdracht

Je hebt een PHP-pagina met een formulier dat soms geen gegevens doorstuurt.

☐ Maak een prompt voor ChatGPT waarin **jalleen het relevante deel van de code opneemt** en waarin je de AI vraagt om te helpen bij het vinden van de fout.

Houd nog steeds rekening met de **context, details en duidelijkheid**.

In dit voorbeeld zou je makkelijk de hele code kunnen verturen, maar in de echte wereld heb je veel meer code en daarom is dit een goede oefening in het isoleren van code die relevant is.

Tip: je kan ook andersom redeneren en alle code waarvan je zeker weet die goed is of niets met het formulier te maken heeft weghaald.

```
<?php
$bericht = "";
$success = false;

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    if (!empty($_POST["naam"])) {
        $naam = $_POST["naam"];
        $bericht = "Hallo, " . htmlspecialchars($naam) . "!";
        $success = true;
    } else {
        $bericht = "Je moet je naam invullen.";
    }
}

?>
```

```
<!DOCTYPE html>
```



```

<html lang="nl">
<head>
  <meta charset="UTF-8">
  <title>Contactpagina</title>
  <style>
    body { font-family: Arial; background-color: #f0f0f0; margin: 2rem; }
    .container { background: white; padding: 2rem; border-radius: 8px; max-width: 600px; margin: auto; }
    .error { color: red; }
    .success { color: green; }
    footer { margin-top: 4rem; font-size: 0.8em; text-align: center; color: #666; }
  </style>
  <script>
    function checkForm() {
      const naam = document.getElementById("naam").value;
      if (naam.trim() === "") {
        alert("Naam is verplicht!");
        return false;
      }
      return true;
    }
  </script>
</head>
<body>

<div class="container">
  <h1>Neem contact met ons op</h1>

  <form method="get" action="" onsubmit="return checkForm();">
    <label for="naam">Naam:</label><br>
    <input type="text" id="naam" name="naam" placeholder="Vul je naam in"><br><br>
    <label for="email">E-mail:</label><br>
    <input type="email" id="email" name="email" placeholder="voorbeeld@domein.nl"><br><br>
    <label for="vraag">Je vraag:</label><br>
    <textarea id="vraag" name="vraag"></textarea><br><br>
    <button type="submit">Verzenden</button>
  </form>

  <?php if ($bericht): ?>
    <p class="<?= $success ? 'success' : 'error' ?>"><?= $bericht ?></p>

```

```
<?php endif; ?>
</div>

<footer>
  &copy; 2025 Webdev Company | <a href="#">Privacybeleid</a>
</footer>

</body>
</html>
```

Inleveren

De prompt die het probleem uiteindelijk vond en waarin **alleen** de relevante code staat.

De prompt voldoet tevens minimaal aan de basis kenmerken van een prompt: **context, details en duidelijkheid**.

2, Provide lists in bullet points

Uitleg

Als je meerdere dingen van een AI vraagt, is het belangrijk om structuur aan te brengen in je prompt. Door **bullet points of genummerde lijsten** te gebruiken:

- Maak je je prompt overzichtelijk.
- Zorg je ervoor dat de AI geen onderdelen vergeet.
- Help je jezelf om duidelijk te verwoorden wat je wilt.

Bullet points zijn dus niet alleen netjes — ze zijn slim.

Opdracht

Hieronder zie je een afbeelding van een eenvoudige webpagina met meerdere onderdelen.

Het plaatje kan je hier downloaden: [Berglandschap.png](#).

image.png and or type unknown

1. **Bestudeer de afbeelding goed.** Wat zie je allemaal op de pagina?
2. **Bedenk wat je aan de AI zou vragen om precies deze pagina te laten maken.**
3. **Noteer minstens 4 onderdelen van de pagina als bullet points** in je prompt.

Inleveren

1. Prompt
2. Resultaat in code
3. Schermafdruck van resultaat

3. Provide the order if you ask for multiple tasks

Uitleg

Als je aan de AI vraagt om **meerdere dingen tegelijk te doen**, dan is het belangrijk dat je duidelijk maakt in **welke volgorde** dat moet gebeuren.

AI volgt jouw instructies letterlijk — dus als de volgorde onduidelijk is, krijg je soms een rommelig of onvolledig resultaat.

Door een logische **nummering** of expliciete volgorde te geven, help je de AI om stapsgewijs te werken.

Voorbeeld (vaag):

Maak een contactpagina met HTML en verwerk de gegevens met PHP.

Voorbeeld (duidelijk):

1. Maak eerst een HTML-pagina met een formulier (naam + e-mail)
2. Voeg daarna de PHP-code toe die de ingevulde gegevens toont

Opdracht

Je wilt de AI vragen om je te helpen met het maken van een eenvoudige contactpagina.

Die moet bestaan uit:

- Een HTML-formulier waarin je je naam en e-mailadres kunt invullen
 - Een PHP-script dat de gegevens verwerkt en netjes op het scherm toont
1. Schrijf een prompt waarin je deze twee onderdelen vraagt in de juiste **volgorde**.
 2. Gebruik een **genummerde lijst** of maak duidelijk met woorden wat “eerst” en “daarna” moet gebeuren.
 3. Houd rekening met de **context, details en duidelijkheid**.

Inleveren

1. Je volledige prompt
2. De gegenereerde HTML + PHP code
3. Een screenshot van het resultaat in de browser

Prompt Engineering 3