Programma 2025/2026

Ga naar: Startjaar 2026

jaar 1

Periode 1		Periode 2		Periode 3		Periode 4	
Blok 1	Blok 2	Blok 3	Blok 4	Blok 5	Blok 6	Blok 7	Blok 8
Scratch -> Python	Retro Gaming Python	Front End CSS	Back End PHP	JavaScript	Databases	Deployment en FrameWorks	Security en OOP
Scratch 1	Vallende Stenen	HTML Phoenix	Intro Web XAMPP	Prompt Engineering 2	DB Design	Prompt Engineering 3	Cyber Security 1
Scratch 2	Snake	CSS Phoenix	Prompt Engineering 1	JS 2 (DOM)	SQL	Deployment	Cyber Security 2
naar Python	Intro AI	Intro JS Phoenix	PHP 2	JS 3 (Dom)	PDO	CMS	OOP
Pak de Kaas	Snake Challenge	Portfolio Challenge	PHP Challenge	JS Challenge	Crud Challenge	Yii Intro	OOP Challenge
Kennis-Check Blok 1	Kennis-Check Blok 2	Kennis-Check Blok 3	Kennis-Check Blok 4	Kennis-Check Blok 5	Kennis-Check Blok 6	Kennis-Check Blok 7	Kennis-Check Blok 8

jaar 2

Periode 1		Periode 2		Periode 3		Periode 4	
Blok 9	Blok 10	Blok 11	Blok 12	Stage	Stage	Stage	Stage

PHP -> Laravel	React en Laravel	GIT en Python	Project Uitvoering Intro Kerntaken	Stage en Examenvoor-bereiding	Stage en Examenvoor-bereiding	Stage en Examenvoor-bereiding	Stage en Examenvoor-bereiding
PHP OOP (herhaling)	Larave 3 Auth	GIT2	SCRUM				
GIT 1	Intro React	Python 1	Reflectie				
Laravel 1 Basic	Laravel 4 React	Python 2	Presenteren				
Laravel 2 Packages	Laravel 5 Advanced	AI Tools	Overleggen	Examen-portfolio	Examen-portfolio	Examen-portfolio	Examen-portfolio
Kennis Check Blk 9	Kennis Check Blk 10	Kennis Check Blk 11	Kennis Check Blk 12				
SMART Technology	SMART Technology	SMART Technology	SMART Technology	Keuzedeel Smart Technolgy	Keuzedeel Smart Technolgy	Keuzedeel Smart Technolgy	Keuzedeel Smart Technolgy

jaar 1 - toelichting

Blok 1 - van blok naar tekst

(totaal ongeveer 9-10 uur)

▶ *Scratch 1 -(block-based programmeren)*

◫ Wat ga je leren?

In **Scratch 1** bouw je stap voor stap je eerste game – een doolhofspel:

- Je ontdekt de **interface**: waar je blokken, sprites en kleuren checkt .

- Je leert een sprite te laten **bewegen** met pijltjestoetsen of de groene vlag .
- Daarna voeg je **loops** toe: code die automatisch herhaalt (handig voor lopen, dansen, geluidjes) .
- Met **if-dan** logica voeg je slimme interactie toe: bijvoorbeeld, als je een muur raakt, gebeurt er iets .
- Tot slot leer je variabelen voor je **score**, zodat je merkt dat je echt iets zelf aanstuurt – superpower!

Waarom dit vet is

- **Je maakt écht iets**: geen saaie theorie, maar een game die je zelf bouwt.
- **Visueel programmeren** maakt het laagdrempelig – je ziet meteen wat er gebeurt.
- **Logisch denkwerk** wordt je nieuwe skill: je splits een probleem in stukjes, bedenkt hoe je dat moet oplossen, en bouwt het.
- **Creatief bezig**: je kiest sprites, kleuren, bewegingen en geluidjes – alles om jouw game uniek te maken.

Het is in totaal zo'n ±145 minuten aan opdrachten, met video's, uitleg en toffe challenges

Onderdeel	Geschatte tijd (in minuten)
Uitleg en oriëntatie op interface	20
Opdracht 1: Sprite laten bewegen	15
Opdracht 2: Dansende sprite	20
Opdracht 3: Geluid toevoegen	15
Opdracht 4: Herhalen gebruiken	20
Opdracht 5: Start met groene vlag	10
Opdracht 6: Eigen sprite gebruiken	20
Opdracht 7: Sprite reageert op toetsen	25
Totaal geschat	**145 minuten**

▶ *Scratch 2 (block-based programmeren)*

□□ Bouw je eigen platformspel – net als Mario!

In deze module leer je stap voor stap hoe je je **eigen platformspel maakt in Scratch** – een beetje zoals Super Mario, maar dan met jouw eigen ideeën en stijl! □□□□

Je begint eenvoudig: een sprite die kan **lopen en springen**. Maar al snel voeg je **muren, platforms en obstakels** toe. Daarna wordt het steeds leuker (en spannender): je leert hoe je **tegen monsters vecht**, hoe je **kunt springen tegen muren**, en hoe je **levels uitdagender maakt**.

Wat je gaat doen

- Sprite laten **bewegen** en **springen**
- Platforms bouwen waar je op kunt staan
- Obstakels toevoegen die je moet **vermijden**
- **Monsters** laten verschijnen (en verslaan!)
- Effecten maken zoals schade of score
- Bij elke stap krijg je duidelijke uitleg met een **video**
- Je maakt screenshots van je werk om in te leveren en punten te verdienen

Waarom dit tof is

- Je leert **hoe games echt werken**
- Je oefent met **logisch denken en programmeren**
- Je gebruikt je **creativiteit** om het spel helemaal van jou te maken
- Je ziet **direct resultaat** van wat je bouwt – dat motiveert enorm!

Deze module kost in totaal ongeveer 160 minuten

Onderdeel	Geschatte tijd (in minuten)
Uitleg random / willekeurige beweging	15
Opdracht: sprite beweegt random	20
Botsing detecteren tussen sprites	20

Onderdeel	Geschatte tijd (in minuten)
Opdracht: reactie op botsing	20
Variabele uitleg + maken van score	20
Score zichtbaar maken op scherm	10
Eindopdracht: game maken met score en botsing	40
Testen / verbeteren van eigen game	15
Totaal geschat	**160 minuten**

▶ *Van Scratch naar Python*

In deze module stap je over van Scratch naar échte code in Python. Geen blokjes meer, maar zelf typen! Je leert de basis zoals **inspringen**, **if-statements**, en **loops**, en ontdekt hoe je een stip laat bewegen, stuiteren of zelfs een spiraal laat tekenen.

Alles gebeurt stap voor stap, met herkenbare voorbeelden uit Scratch. Je werkt in een makkelijke programmeeromgeving (Thonny), zodat je je kunt focussen op het leren.

Aan het eind kijk je terug: wat heb je geleerd, wat ging goed en wat kun je nog verbeteren?

Kortom: een supertoffe en toegankelijke eerste stap naar écht programmeren! □□□□

Deze module kost ongeveer 140 minuten

Onderdeel	Geschatte tijd (in minuten)
Uitleg overgang Scratch → Python	15
Eerste turtle-voorbeeld tekenen	15
Uitleg over regels code en syntax	20
Opdracht: vierkant tekenen	15
Uitleg en gebruik van `for`-loop	20
Opdracht: herhaling met `for`-loop	20
Opdracht: maak een patroon met meerdere loops	25
Reflectie of bespreking	10
Totaal geschat	**140 minuten**

▶ *Pak de Kaas (eenvoudig spel in Python)*

In deze module leer je stap voor stap hoe je in Python een eigen game maakt. Je begint met het tonen van afbeeldingen (sprites) en leert daarna hoe je een speler laat bewegen en botsingen detecteert. Daarna voeg je extra's toe zoals **willekeurige bewegingen**, een **score** en zelfs een **tijdslimiet**.

Je oefent met belangrijke programmeerconcepten zoals **if-statements**, **loops** en **variabelen**. Ook leer je hoe je grote problemen opsplitst in kleine stukjes code – precies zoals echte programmeurs dat doen.

Je leert dus niet alleen games maken, maar ook **slim nadenken in stappen**. Zo word je steeds handiger in het schrijven én begrijpen van code. □□□□

Deze module kost ongeveer 120 minuten

Onderdeel	Geschatte tijd (in minuten)
Uitleg overgang Scratch → Python	15
Eerste turtle-voorbeeld tekenen	15
Uitleg over regels code en syntax	20
Opdracht: vierkant tekenen	15
Uitleg en gebruik van `for`-loop	20
Opdracht: herhaling met `for`-loop	20
Opdracht: maak een patroon met meerdere loops	25
Reflectie of bespreking	10
Totaal geschat	**140 minuten**

Blok 2 - spelletjes in Python

(totaal ongeveer 8 - 9 uur)

▶ *Vallende Stenen (spel Python)*

□□ Tover je kennis om in een eigen game

In deze module ga je verder met wat je al hebt geleerd in Scratch en "Van Scratch naar Python". Je gebruikt **variabelen**, **if-statements** en **for-loops** om een spel te maken waarin objecten vallen en botsingen plaatsvinden.

Je oefent met het **lezen en aanpassen van code**, zodat je snapt hoe iets werkt én hoe je het kunt verbeteren. Denk aan: voorkomen dat je speler uit beeld vliegt, of meerdere objecten tegelijk laten bewegen.

Aan het eind laat je jouw **creativiteit los**: voeg zelf nieuwe functies toe zoals geluiden, power-ups of een andere look. Zo maak je het spel écht van jou! □□□□

Deze module kost ongeveer 120 minuten

Onderdeel	Tijd (in minuten)
Spelconcept begrijpen	10
Speler-sprite tekenen + toetsen besturen	20
Steen laten vallen vanaf random plek	25
Herhalen van de val (loop gebruiken)	15
Botsing detecteren	20
Spel laten stoppen of levens aanpassen	15
Testen en bijschaven	15
(Optioneel) Extra's zoals score of geluid	20
Totaal zonder extra's	**120 minuten**
Totaal met extra's	**140 minuten**

▶ *Snake (spel Python)*

□□ Bouw je eigen Snake-game in Python

In deze module ga je aan de slag met het maken van een echte **Snake-game** met Pygame Zero. Je herhaalt bekende concepten, maar de opdrachten zijn wat pittiger en de code wordt langer.

Je leert nieuwe dingen, zoals **hoe de slang groeit** als hij iets eet – een tricky stukje logica! Aan het eind krijg je een opdracht waarbij je goed moet **nadenken over een game-probleem**, een mooie voorbereiding op later werk met AI en prompt engineering.

Kortom: een leuke maar uitdagende module waarin je laat zien wat je al kunt, én nieuwe skills leert! □□□□

Deze module kost ongeveer 160 minuten

Onderdeel	Tijd (in minuten)

Spelidee en opdracht begrijpen	10
Basisbesturing + beweging slang	25
Eten plaatsen en detectie bij aanraking	20
Snake laten groeien na eten	25
Zelfbotsing implementeren	25
Score bijhouden	15
Game-over logica (botsing muur/zelf)	15
Testen, debuggen en verbeteren	20
(Optioneel) Extra functies	20
Totaal zonder extra's	**155 minuten**
Totaal met extra's	**175 minuten**

▶ *Intro AI (wat is AI en wat zijn voor- en nadelen?)*

□□ Wat is AI – en hoe gebruik je het slim?

In deze korte module ontdek je het verschil tussen **klassiek programmeren** (met vaste regels) en **AI**, die juist **leert van data**. Je leert welke soorten AI er zijn – zoals **classificatie** en **voorspellen** – en je denkt na over de **voordelen én risico's**van AI.

Ook maak je kennis met **prompt engineering**: hoe stel je goede vragen aan AI om het nuttig en eerlijk te gebruiken. Deze les is een opwarmertje voor wat nog komt én een voorbereiding om AI slim in te zetten bij je Snake Challenge. □□⚙

Deze module kost ongeveer 70 minuten

Onderdeel	Tijd (in minuten)
Klassikale uitleg en voorbeelden	20–30
Video of demonstratie (indien aanwezig)	10
Vragen beantwoorden en bespreken	20–30
Reflectie of groepsdiscussie	15
(Optioneel) korte opdracht of Kahoot	10–15
Totaal	**60–90 minuten**

▶ *Snake Challenge (Python/AI challenge)*

Studenten woerden in deze **challange** uitgedaagd om een **eigen versie** van hun Snake game te maken.

Deze module duurt ongeveer 120 minuten

Activiteit	Tijd (in minuten)
Ideeën bedenken (eventueel klassikaal)	15
Eén uitbreiding kiezen en implementeren	30–45
Tweede uitbreiding maken	30–45
Testen, verbeteren en eventueel presenteren	20–30
Totaal geschat	**100–135 minuten**

Blok 3 (web front-end)

(totaal ongeveer 8-9 uur)

▶ *HTML / Phoenix*

In deze module leer je stap voor stap hoe HTML werkt. Elke les behandelt een duidelijk onderwerp, zoals de structuur van een webpagina, tekst opmaken, afbeeldingen en links toevoegen, lijsten en tabellen maken, en formulieren bouwen.

Je werkt in Phoenix Code, een makkelijke editor in je browser waar je direct kunt zien wat je maakt.

Door de opdrachten leer je niet alleen losse onderdelen, maar werk je toe naar een eigen mini-project: een persoonlijke homepage die je helemaal zelf bouwt met wat je hebt geleerd.

Deze module kost ongeveer 100 minuten

Onderdeel	Tijd (in minuten)
Introductie HTML + structuur uitleg	15
Voorbeelden tags + experimenteren	25
Werken met Phoenix (inloggen, omgeving leren)	10
Opdracht: eigen pagina maken	30–40
Testen en bespreken	10–15
Totaal geschat	**90–105 minuten**

▶ CSS / Phoenix

In deze module ontdek je wat CSS is en waarom het onmisbaar is om je webpagina er mooi en overzichtelijk uit te laten zien.

Je leert hoe je kleuren, lettertypes, marges, uitlijning en achtergronden aanpast, en hoe je met het box-model werkt (margin, padding, border). Ook selectors zoals element, class en ID komen aan bod, zodat je precies weet hoe je onderdelen van je site kunt stylen. Met Flexbox leer je moderne lay-outs maken die netjes meeschalen en mooi uitgelijnd zijn.

□□ Het einddoel is dat je zelf goed uitziende, strak opgebouwde webpagina's kunt maken met jouw eigen stijl.

Deze module kost ongeveer 100 minuten

Onderdeel	Tijd (in minuten)
Uitleg wat CSS is en hoe het werkt	15
CSS-syntaxis + voorbeelden toepassen	25
Werken met selectors	10
Opdracht: bestaande HTML-pagina stylen	30–40
Experimenteren en verbeteren	10–15
Totaal geschat	**90–105 minuten**

▶ Intro JS / Phoenix

□□ In deze module duik je in de wereld van JavaScript – de taal die websites tot leven brengt.

Je leert hoe je met variabelen, functies en if/else-logica slimme en interactieve onderdelen maakt. Je gaat meteen praktisch aan de slag in de browserconsole en met HTML-bestanden, zodat je direct ziet wat jouw code doet.Laat je pagina reageren op wat de gebruiker doet, verwerk invoer uit formulieren en bouw je eigen berekeningen.

Met heldere uitleg, leuke opdrachten en reflectievragen leer je stap voor stap hoe jij een simpele website verandert in een echte interactieve webapp.

Deze module kost ongeveer 80 minuten

Onderdeel	Tijd (in minuten)

Uitleg wat JavaScript is en hoe het werkt	10–15
Console openen en verkennen	10
Voorbeelden doornemen en uitvoeren	20
Opdracht: zelf experimenteren met `console.log`	25–30
Klassikale bespreking of korte quiz	10
Totaal geschat	**75–85 minuten**

▶ Snake *Challenge*

De portfolio Challenge combineert persoonlijk profileren met web design en daagt de student uit een eigen portfolio te maken.

Deze module kost ongeeer 150 minute.

Onderdeel	Tijd (in minuten)
Opdracht lezen en onderwerp kiezen	15
Schets of opzet maken	15–20
HTML schrijven (structuur opzetten)	30–45
CSS toevoegen en verbeteren	30–45
(Optioneel) JavaScript toevoegen	30
Testen en bijschaven	20–30
Reflectie/samenvatting/afronding	10–15
Totaal geschat (zonder JS)	**120–150 minuten**
Totaal geschat (met JS)	**150–180 minuten**

Blok 4 (web back-end)

(ongeveer 7-8 uur)

▶ *PHP - XAMPP*

□□ In deze module leer je wat het verschil is tussen frontend (wat je ziet) en backend (wat er op de server gebeurt) – een belangrijke eerste stap richting echte webontwikkeling.

Je installeert tools als XAMPP en Visual Studio Code om zelf lokaal met PHP aan de slag te gaan en je code te testen. Je maakt je eerste PHP-pagina's, werkt met variabelen en teksten, en bouwt interactieve onderdelen zoals formulieren en een kortingscalculator.

Alle oefeningen zijn praktisch en helder opgebouwd, zodat je direct merkt wat je doet én waarom. Zo leer je echt webapplicaties bouwen, van de basis tot iets dat werkt!

Deze module kost ongeveer 90 minuten

Onderdeel	Tijd (in minuten)
Uitleg over wat PHP is	10–15
Eerste regels code schrijven en testen	15
Uitleg over `echo` en variabelen	15
Opdrachten uitvoeren (3 à 4 opdrachten)	30–40
Testen, fouten zoeken en verbeteren	10–15
Totaal geschat	**80–100 minuten**

▶ *Prompt Enineering*

De module, "Prompt Engineering 1", dient als een handleiding voor het effectief communiceren met AI-modellen zoals ChatGPT.

In deze module leer je hoe je beter kunt communiceren met AI, zoals ChatGPT. Je ontdekt dat hoe duidelijker en specifieker je vraag is, hoe beter het antwoord dat je krijgt.

Je werkt met zes belangrijke elementen van een goede prompt: context, details, duidelijkheid, doel, vorm en toon.

✍ Door praktische voorbeelden en opdrachten leer je hoe je deze elementen toepast om slimme, gerichte vragen te stellen. Zo haal je meer uit AI en krijg je antwoorden die echt passen bij wat jij nodig hebt – handig voor school, projecten en later werk!

De module kost ongeveer 90 minuten

Onderdeel	Tijd (in minuten)
Introductie prompt engineering	10–15
Voorbeelden en klassikale analyse	15
Tips bespreken	10

Onderdeel	Tijd (in minuten)
Opdrachten: prompts verbeteren + AI testen	30–40
Reflectie of klassikale bespreking	10–15
Totaal geschat	**80–95 minuten**

▶ PHP 2

In deze module leer je hoe je met PHP de achterkant van een website bouwt – alles wat de gebruiker niet ziet, maar wel gebeurt op de server.

Je leert veilig gegevens versturen via GET en POST, en hoe je je code slim organiseert met include en require. Met arrays en loops verwerk je makkelijk data, en met functies maak je je code herbruikbaar en overzichtelijk. Ook werk je met datum en tijd, en ontdek je hoe je sessies gebruikt om bijvoorbeeld een simpele login te maken met gebruikersgegevens.

Elke les bevat uitleg, opdrachten en reflectievragen waarmee je stap voor stap werkt aan je skills als webdeveloper.

De module kost ongeveer 90 minuten

Onderdeel	Tijd (in minuten)
Uitleg over formulieren in HTML	15
Uitleg over `$_GET` en hoe data via URL binnenkomt	15
Eerste voorbeeld doornemen en uitvoeren	20
Opdracht uitvoeren: formulier maken + verwerken	25–30
Testen, fouten oplossen, extra uitdaging	10–15
Totaal geschat	**85–95 minuten**

▶ PHP Challenge

In deze PHP Challenge laat je zien wat je hebt geleerd door zelf een mini-website te bouwen met echte backend-functies.

Je kiest zelf wat je maakt – bijvoorbeeld een persoonlijke site, quiz of simpele webshop – zolang het project maar voldoet aan de technische eisen.Denk aan meerdere PHP-pagina’s, het gebruik van formulieren, sessies, functies, arrays en loops.

Aan het eind laat je jouw project zien en reflecteer je op wat je hebt geleerd. Zo toon je niet alleen je resultaat, maar ook je groei als developer.

De module kost ongeveer 110 minuten

Onderdeel	Tijd (in minuten)
Uitleg opdracht en doelen begrijpen	10–15
Zelf quizvragen bedenken en formulier opzetten	20–30
PHP-code schrijven om antwoorden te controleren	25–30
Score berekenen en tonen	15
Testen, debuggen en verbeteren	15–20
(Optioneel) Extra’s zoals feedback of styling	15–20
Totaal geschat	**100–130 minu**

Blok 5 (web back-end 2)

▶ Prompt Engineering 2

□□ Deze module bouwt voort op wat je al weet over prompt engineering en neemt je mee naar **geavanceerde technieken**die je helpen AI écht slim in te zetten. Je werkt aan **zes praktische strategieën** die niet alleen je prompts verbeteren, maar ook je probleemoplossend denken scherpen

Met per techniek handige uitleg, voorbeeldprompts en opdrachten leer je niet alleen AI te sturen, maar ook zelf kritisch na te denken – handig voor complexe taken zoals prompt engineering.

Het resultaat? Jij weet hoe je met slimme prompts AI antwoorden krijgt die aansluiten op jouw situatie en leerbehoefte. Deze module is de perfecte springplank voor je Snake Challenge en verdere AI-projecten!

▶ JavaScript 2 (DOM)

In deze module leer je hoe je met JavaScript de **DOM** (Document Object Model) kunt aanpassen om je webpagina interactiever te maken.

e leert HTML-elementen **selecteren** (zoals met getElementById of querySelectorAll) en de **inhoud en stijl** ervan aanpassen via JavaScript. Je oefent met **events**, zoals klikken en muisbewegingen, en leert reageren op gebruikersinput met addEventListener Je maakt nieuwe elementen aan (

createElement) en voegt ze toe of verwijdert ze in de DOM met appendChild, remove() of event.target. Met classList.toggle voeg je eenvoudig styling classes toe of haal je ze weg om interactie visueel te maken.

Door stap-voor-stap opdrachten bouw je van basisselectie tot volledige interactiviteit – van tekst aanpassen, elementen markeren en dynamisch toevoegen tot verwijderen en style toggles. Zo leer je hoe je JavaScript écht kan inzetten om je pagina levendiger te maken.

▶ Javascript 3 (DOM)

In deze module duik je dieper in het **DOM**, een model dat jouw webpagina omzet in objecten waar JavaScript mee kan werken – de brug tussen HTML en interactieve code. Je gaat creatief aan de slag met het **aanmaken, toevoegen en verwijderen van elementen** (createElement, appendChild, remove()), en je leert hoe je attributen instelt of classes toggle't voor dynamische effectjes Met opdrachten bouw je van basisselectie tot volledige interactiviteit: je markeert dingen, voegt nieuwe componenten toe of verwijdert ze, en maakt pagina's echt levend

Het resultaat? Je krijgt niet alleen inzicht in hoe de DOM werkt, maar je krijgt ook de tools om je webpagina's interactief te maken op een manier die echt gaat leven. Veel plezier met knutselen!

▶ JavaScript Challenge

In deze JavaScript Challenge bouw je een volledig interactieve webapp om je front-end skills te showen.

Je laat de DOM tot leven komen met events, functies en dynamische updates op basis van user-input. Kies je eigen concept—todo-app, quiz of mini-game—zolang je CRUD-acties en strakke UX gebruikt.

Presenteer je app, reflecteer op je code en bewijs dat jij JavaScript onder de knie hebt!

Blok 6 (Database / PDO)

▶ *Database*

Deze module gaat over **databaseontwerp**. Het leidt de student stapsgewijs door de fundamentele concepten van databases, beginnend met **wat een database is** en waarom deze nodig is voor het organiseren van gegevens. De module introduceert cruciale terminologieën zoals **entiteiten en attributen**, legt het belang uit van een **Primary Key (PK)** voor unieke identificatie, en behandelt **1:N-relaties en Foreign Keys (FKs)** om tabellen te verbinden. Tot slot wordt dieper ingegaan op **datatypen**, de **modellering van realistische scenario's**, en de complexere **N:N-relaties**, waarbij het creëren van tussentabellen wordt uitgelegd voor optimale datastructuren.

▶ *SQL*

In deze module ga je aan de slag met **SQL en databases**, zodat je leert hoe je data effectief opslaat, opvraagt en beheert.

Je start met de basis: wat SQL is en waarom het essentieel is voor het bouwen van dynamische webapplicaties. Je leert hoe je tabellen maakt en vult, hoe je specifieke gegevens ophaalt met SELECT, en hoe je data aanpast met INSERT, UPDATE en DELETE. Relaties tussen tabellen komen aan bod: je maakt query's met **JOINs** om gegevens uit meerdere tabellen te combineren. Ook verdiep je je in belangrijke SQL-technieken zoals **GROUP BY**, **WHERE**, **ORDER BY**, en eenvoudige **aggregatiefuncties** zoals COUNT() en SUM().

Door praktische oefeningen leer je hoe je SQL gebruikt om echte businessvragen te beantwoorden: van rapportages maken tot data-analyse in je database.

Het doel is dat je na deze module zelfstandig databasequery's schrijft om informatiesystemen te bouwen die écht werken – een onmisbare skill voor elke developer!

▶ *PDO*

In deze module leer je werken met **PDO**, de moderne en veilige manier om met databases te praten in PHP. Je maakt verbinding met een database, voert queries uit en gebruikt **prepared statements** om SQL-injecties te voorkomen. Je oefent met data ophalen (fetch), invoegen, updaten en verwijderen – allemaal op een nette en herbruikbare manier. Ook kijk je hoe je PDO slim inzet in je eigen project, bijvoorbeeld met een klasse of in combinatie met forms.

Na deze module schrijf je zelfverzekerd veilige en nette databasecode – klaar voor het echte werk!

▶ CRUD Challenges

In deze **CRUD Challenge** laat je zien wat je met PHP en PDO geleerd hebt door een werkende mini-website te bouwen – net als een echte developer.

Je kiest zelf een project, bijvoorbeeld een persoonlijke site, quiz of webshop, en bouwt dat in meerdere PHP-pagina's. Daarbij pas je toe wat geleerd is: **formulieren**, **arrays**, **loops**, **functies** en **sessies** (voor inlog of winkelwagen).

Blok 7 (PHP Frameworks)

▶ Prompt Engineering 3

Prompt Engineering 3 is een gedetailleerde handleiding voor **geavanceerde prompt-technieken** binnen softwareontwikkeling, specifiek gericht op het optimaliseren van interacties met AI-modellen. Het document bouwt voort op eerdere modules over basale promptkenmerken zoals **context, details, duidelijkheid, doelgerichtheid, vorm en toon**, en introduceert zes geavanceerde methoden. Deze omvatten het **gebruiken van rollen (role prompting)**, het vragen om **alternatieven of variaties**, **iteratief verbeteren** van antwoorden, het toepassen van **prompt templates**, het stimuleren van **zelfvragen (self-questioning)** door de AI, en het benutten van **kritiek als leermoment (prompt debugging)**. Het doel is gebruikers te leren hoe ze effectievere en relevantere output van AI kunnen verkrijgen door middel van strategische en kritische promptformulering.

▶ Deployement

Deze module leidt studenten stap voor stap door het proces van het online zetten van een website of webapplicatie. Belangrijke onderwerpen zijn het verkrijgen van **toegang tot PLESK**, een beheerpaneel voor webhosting, het **beveiligen van een domein met een SSL-certificaat** voor een veilige verbinding, en het **aanmaken van een FTP-account** om bestanden naar de server te uploaden. Verder wordt uitgelegd hoe een **database wordt aangemaakt** en hoe een zelfgebouwde webapplicatie **live wordt geïnstalleerd**, inclusief de noodzaak om databaseverbindingen aan te passen voor de live-omgeving.

▶ CMS

Wordpress is het meest gebruikte Content Management Systeem. Meer dan 43% van alle websites op internet draait op WordPress (stand juni 2025). In deze module leer je Wordpress installeren en daarnaast leer je alle basisconcepten van Wordpress.

▶ Yii Intro

Deze module introduceert de basisprincipes van **webapplicatieontwikkeling** met behulp van het **Yii2 PHP-framework**. De tekst gidst de student door de installatie van Yii en essentiële tools zoals Composer en XAMPP, en demonstreert vervolgens hoe een **CRUD-applicatie** (Create, Read, Update, Delete) snel kan worden gegenereerd voor databasebeheer. Belangrijke concepten zoals het **MVC-architectuurpatroon** (Model-View-Controller) en **routing** worden uitgelegd, waarbij de structuur van een webapplicatie en de manier waarop URL's worden afgehandeld duidelijk worden gemaakt.

Tot slot wordt dieper ingegaan op het aanpassen van de specifieke Yii-gebruikersinterface via de **Gridview-widget**, waardoor de student leert hoe kolommen, labels en inhoud dynamisch kunnen worden beheerd.

Blok 8

▶ Cyber Security 1

"*Deze lessen zijn ontworpen om studenten niet alleen theoretische kennis bij te brengen, maar ook praktische vaardigheden en een kritische houding ten aanzien van digitale veiligheid.*"

In deze module stap je in de wereld van **cybersecurity** – een wereld waarin hackers, virussen en datadieven dagelijks op de loer liggen. Maar geen zorgen: jij leert hoe je je daartegen kunt verdedigen.

We beginnen met een duidelijke uitleg van de **grootste gevaren op internet**, zoals **phishing** en **malware**. Je ontdekt hoe eenvoudig het soms is om iemand te misleiden – en hoe jij dat dus kunt herkennen én voorkomen.

Daarna duiken we in de techniek achter **veilige websites**: wat doet dat slotje in je browser precies? Wat is **HTTPS**, en waarom zijn **SSL-certificaten** belangrijk? Je leert hoe versleuteling werkt, met zowel **symmetrische als asymmetrische encryptie**.

We laten je zien waarom wachtwoorden alleen hashen **niet genoeg is** – en waarom een beetje “salt” het verschil maakt tussen veilig en kwetsbaar. Ook onderwerpen als **Rainbow Tables** en **Brute Force-aanvallen** komen voorbij, mét strategieën om ze tegen te houden.

▶ Cyber Security 2

In **Cyber Security 2** ga je een stap verder dan de basis en leer je hoe je als software developer de digitale wereld echt kunt beschermen. Je verdiept je in het cruciale verschil tussen **risico** (de kans dat iets gebeurt) en **impact** (de gevolgen daarvan), waardoor je leert welke beveiligingsmaatregelen de hoogste prioriteit verdienen.
Deze module maakt je een expert in het herkennen en stoppen van geavanceerde web-aanvallen zoals **SQL-injectie, Cross-Site Scripting (XSS) en Cross-Site Request Forgery (CSRF)**. Door te leren **denken als een hacker**, ontdek je zwakke plekken in je eigen code voordat anderen er misbruik van maken.
Je maakt kennis met de professionele standaard van de **OWASP Top 10** en leert hoe je met **Responsible Disclosure** op een ethische en verantwoordelijke manier kunt bijdragen aan digitale veiligheid. Aan het eind van de module breng je al deze kennis samen om je eigen webapplicatie volledig te analyseren en te beveiligen, een **essentiële vaardigheid** die jou onderscheidt als een professionele en betrouwbare developer

▶ OOP

In deze module leer je het verschil tussen procedureel programmeren (zeg maar 'gewoon' programmeren) en **OOP**.

Deze module is een voorbereiding op het leren omgaan met Frameworks zoals Yii2, Laravel, Flask, etc. etc.

Met **OOP** kun je code beter **organiseren**, deze **herbruiken** en je programma makkelijker uitbreidbaar maken. Je leert complexe systemen op te splitsen in beheersbare blokken met behulp van concepten als **klassen**, **objecten**, **properties** en **methods**.

Constructors zorgen ervoor dat je automatisch waarden kunt toekennen bij het aanmaken van objecten, wat efficiëntie bevordert. Bovendien biedt **encapsulation** de mogelijkheid om data te beschermen, waardoor fouten worden voorkomen en je gecontroleerde toegang tot eigenschappen krijgt via **getters** en **setters**.

▶ OOP Challenge

De **OOP Challenge** is een fantastische kans om jouw vaardigheden in **objectgeoriënteerd programmeren** (OOP) naar een hoger niveau te tillen en direct toe te passen in de praktijk. Je combineert je **Cyber Security** vaardigheden met de **OOP** vaardigheden.

Je hebt de **keuze** uit twee boeiende projecten; Je kunt een login systeem maken of je kunt een Complete Mobiele ToDo manager maken.

--

Versnellen

Versnellen: https://www.roc.ovh/books/overig-notes/page/versnellen

Kennis-Check

Kennis-Checks: https://www.roc.ovh/books/overig-notes/page/kennis-checks

---

Revision #135
Created 2025-05-28 11:11:04 UTC by Max
Updated 2026-11-17 22:20:53 UTC by Max