

# Overig Onderwijs

Losse pagina's tbv Onderwijs

- [Kerntaak3 - productie](#)
- [Recording](#)
- [Database Design](#)
- [ExamenExec \(in Yii2\)](#)
- [Challenges](#)
- [Rolspeler \(Kerntaak-1\)](#)
- [PHP Forms en Arrays](#)
- [Intake](#)
- [Framework](#)
- [ExamBot](#)
- [Kerntaak 3](#)
- [KT Beoordeling](#)
- [MariaDB](#)
- [ED Status 3de jaars](#)
- [Checklist DB Design \(EN\)](#)
- [Windows Firewall](#)
- [PDG - cheat sheet](#)
- [Database Design \(En\)](#)
- [OOP Eindopdracht](#)

- [Groep 8 - Introductie web site maken](#)
- [Yii installeren in live omgeving](#)
- [Stoplicht](#)
- [10-tallig stelsel](#)
- [Concreet of niet - Versie A](#)
- [Concreet of niet - versie B](#)
- [Extra Opgave DB1](#)
- [JS Challenge - Stappenplan](#)
- [Authenticator app op een nieuwe telefoon activeren.](#)
- [Namen en Alliteraties](#)
- [Kennismakingsspellen](#)
- [\(Cyber\) Security Overview](#)
- [AI en Onderwijs](#)
- [PDO](#)

# Kerntaak3 - productie

## in Productie plaatsen

### Test het in productie nemen van een voorbeeld site, voor je examen.

Nu kan ik je nog helpen, tijdens jouw examen niet meer.

Test het liefst een kleine PHP app die een of een paar regels uit jouw database haalt. Op die manier weet je dat jouw database werkt, en hoe je die moet verbinden met jouw code.

Hieronder wordt beschreven hoe je kunt aanloggen op de productieserver en hoe je files kunt over kopiëren.

## Wachtwoord

Log in op [examen.student.ovh](https://examen.student.ovh) en noteer jouw wachtwoord dat onderaan de pagina staat.

image-1612213107563.png

Jouw userid is jouw studentnummer met een u ervoor.

Bijvoorbeeld u3001206

Je hebt nu een userid en een wachtwoord en dat heb je nodig om jouw site live te zetten.

## SFTP (of ssh)

Je kunt met ssh (rsync) of met sftp files op de server zetten.

Ik zal het gebruik van sftp met WinSCP (<https://winscp.net/eng/download.php>) uitleggen.

image-1612213505897.png

Vul de volgende velden in:

- Host name: `softwaredeveloper.ovh`
- Port nummer: 1221
- User name: `<jouw user naam>`, dus een u gevolgd door jouw studentnummer

- Jou wachtwoord

## Druk op login

image-1612435871580.png

Hier kun je jouw website (files) uploaden. Als je de weg kwijt bent kun je altijd naar deze folder terugkeren door op het home knopje te drukken.

Jouw website staat op:

<username>.softwaredeveloper.ovh.

bijvoorbeeld

u596621.softwaredeveloper.ovh

De index.hp in de ww directory laat jouw standaard website zien, bijvoorbeeld:

image-1612214065223.png

Plaats nu met WinSCP jouw website in de(ze) productieomgeving.

## Database

De database kun je benaderen via <http://www.softwaredeveloper.ovh/phpmyadmin/>

Jouw user naam en password zijn hetzelfde als om in te loggen op de server met SFTP.

Via een database import kun je jouw database installeren.

Als je aanlogt met phpmyadmin dan is er een lege database aangemaakt. Deze database staat op dezelfde server als jouw (php) code. Je kunt dus localhost of 127.0.0.1 als server naam gebruiken. Je moet wel jouw database naam en login wijzigen.

## Voorbeeld filmpje applicatie in productieomgeving plaatsen

[https://www.youtube.com/embed/F\\_Avy\\_rFKqM](https://www.youtube.com/embed/F_Avy_rFKqM)

In dit filmpje worden de volgende stappen voorgedaan:

1. Ga naar <https://www.tutorialrepublic.com/php-tutorial/php-mysql-crud-application.php>

2. Maak een nieuwe file config.php en kopieer de inhoud van de config van de website naar deze file.
3. Maak een nieuwe file test.php en kopieer de inhoud van de landing page van de website naar deze file.
4. Maak een test database (naam *test*).
5. Maak één tabel, *employee* in de database (de tabel definitie staat ook op de website).
6. Vul de database met een paar regels (records), bedenk zelf wat waardes.
7. Als het goed is heb je nu een applicatie lokaal op je laptop, controleer of die werkt.
8. Pas de config aan aan de productieomgeving.
9. Plaats de twee files (config.php en test.php) naar productie.
10. Maak een database export en importeer die export in je productie database.
11. Controleer of jouw applicatie in productie werkt.

## Links naar lessen

### Les 1

<https://web.microsoftstream.com/video/def8f57c-2169-4e05-a2af-51510f17d907>

<https://web.microsoftstream.com/video/65767b6e-c932-4db7-aa51-ad806c09bc6f>

### Les 2

<https://web.microsoftstream.com/video/c85a9024-5f01-4a8a-ad5a-8b5b84d18306>

<https://web.microsoftstream.com/video/e278bd8b-7638-4c50-9ab6-598afab030e0>

---

# Recording

## Screen recording MS Stream

Gebruik **Chrome** en ga naar

<https://web.microsoftstream.com>

**Kies menu create, 'Record screen or video'**

image-1611854042252.png

**Kies Screen only en zet de microfoon aan**

image-1611854148552.png

**Druk op start recording**

image-1611854243455.png

**Kies Your Entire Screen**

image-1611854466983.png

De opname begint. Voer nu je presentatie uit.

**Als je klaar met de opname bent druk je op 'next'**

image-1611854590213.png

**Daarna druk je op 'upload to Stream'**

image-1611854654415.png

**Publish jouw video.**

image-1611854706617.png

**Ga naar de video.**

image-1611854747695.png

**Ga naar de video details**

image 1611855368553.png

Geef iedereen (binnen ons ROC) de view rechten of als je dat niet wilt dan mag je de video ook alleen met de examinatoren delen delen.

De examinatoren zijn:

- [m.bisschop@rocva.nl](mailto:m.bisschop@rocva.nl)
- [t.monincx@rocva.nl](mailto:t.monincx@rocva.nl)
- [r.vanpietersom@rocva.nl](mailto:r.vanpietersom@rocva.nl)

image 1611855553052.png

**Druk op Apply**

image 1611855611006.png

**Deel de video.**

image 1611854824210.png

**Zet de link in een document en lever die in Teams in.**

image 1611854847068.png

**Klaar**

--

# Database Design

*In deze les gaan we oefenen met het maken van een datamodel. Binnen software development is het bedenken van een datamodel een belangrijk onderdeel. Het kan worden gezien als het fundament waarop de applicatie wordt gebouwd. Als het datamodel niet goed is dan is het fundament niet juist. Aanpassingen van het datamodel leiden vaak tot ingrijpende en lastige wijzigingen in de code.*

**Een goede applicatie begint bij een goed datamodel!**

## Inleiding

In deze les ga je oefenen met het maken van een datamodel, dit heet een ERD.

Het ERD, het **Entity Relation Diagram**, is een schematische weergave van het datamodel.

Een ERD is een schematische weergave van entiteiten, attributen, relaties en sleutels.

### Entiteiten

Dit zijn dingen, mensen, of gebeurtenissen waarover je iets wilt vastleggen. Bijvoorbeeld *studenten* (mensen) en *ziektemeldingen* (gebeurtenissen).

### Attributen

De attributen zijn de eigenschappen van de entiteiten. Bijvoorbeeld de naam van een student of de datum van een ziektemelding.

### Relaties

Relaties de verbinding tussen de entiteiten. Deze kun je herkennen doordat je een zin kan maken die begint met *iedere* of *elk* en de zin bevat een werkwoord. In de zin moet je ook iets zeggen over de hoeveelheid. Elke relatie kan je van twee kanten beschrijven, voorbeeld:

- ↳ iedere student heeft zich nul. één of meer keren ziek gemeld.
- ↳ iedere ziektemelding hoort bij precies één student

### Sleutels



Slutels zijn attributen die nodig zijn voor om de relatie te maken. Elke entiteit heeft een uniek id. Voor de eenvoud noemen we die altijd id (type integer). Dit heet de primary key. De foreign key is een verwijzing naar een primary key uit een andere entiteit.

# Hoe maak je een ERD?

Elke opgave bestaat uit een verhaaltje dit wordt ook wel een case genoemd. Lees de case goed door en bepaal van welke entiteiten je gegevens wilt vastleggen. Bepaal van alle entiteiten *wat* je wilt vastleggen, dit zijn de attributen. Bepaal van alle attributen het datatype (integer, float, date, time, datetime of boolean). Vervolgens bepaal je de relatie tussen de entiteiten en bepaal je de primary keys en foreign keys.

Dus de stappen zijn:

1. Bepaal van welke entiteiten je gegevens wilt vastleggen in de database.
2. Bepaal van elke entiteit welke gegevens je wilt vastleggen, dit zijn de attributen en bepaal het datatype per attribuut.
3. Bepaal de relaties tussen de entiteiten
4. Bepaal de PK's en FK's.

De datamodellen worden gemaakt in [Lucichart](#), een datamodel kun je in PDF exporteren en op je eigen systeem bewaren.

[image.1603478193579.png](#)

Om een ERD in [Lucichart](#) te maken, maak je een gratis account en zoek je de ERD-template "Database ER Diagram", zie plaatje hierboven.

## Naming conventions

Het database design (het ERD) wordt in het Engels gemaakt. Entiteiten en Attributen worden via de **Snake Case** (in lower case) benoemd, bijvoorbeeld:

Entiteit: user

Attribuut 1: user\_name

Attribuut 2: user\_login\_count

Attribuut 3: last\_login\_date

Attribuut 4: password

Let ook op dat entiteiten **enkelvoud** zijn. Dus bijvoorbeeld *user* en niet *users*, of *leerlingen* niet *leerlingen*.

## Voorbeeld ERD

[image-1603486029048.png](#)

Dit diagram is in Lucichart gemaakt. Herken je de entiteiten en de attributen? Wat kun je van de relatie zeggen? Wat wordt er vastgelegd in deze database? Herken je de foreign key? Waar verwijst de foreign key naar?

## De normalisatie regels

Als je niet in een keer de entiteiten herkent kun je ook alle gegevens in één tabel zetten. Vervolgens ga je deze tabel dan net zo lang splitsen totdat aan de onderstaande regels is voldaan.

Denk je wel alle entiteiten te herkennen, controleer dan ook of je aan alle regels voldoet.

1. Elk veld bevat één gegeven.
2. Gegevens uit één kolom betekenen hetzelfde.
3. Elke regel is uniek.
4. Kolommen mogen niet herhaald worden (veld1, veld2, veld3,...).
5. “zo min mogelijk” dubbele data (strings), aanpassingen hoeft je maar één keer te doen.
6. Aanpassen van gegevens kan door één veld aan te passen. (ook wel; elke table gaat over één ding).

De regels kan je ook samenvatten: ze zorgen ervoor dat: elk veld op elke regel kan veranderen zonder dat je dan ook allemaal andere gegevens moet wijzigen

Als je jouw database model (jouw tabellen) niet aan alle regels voldoen dan moet je gegevens splitsen. Daarvoor heb je drie normalisatie technieken.

## De normalisatie technieken

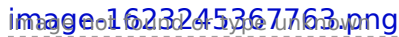
### Kolommen splitsen

Splits een kolom in twee of meer kolommen.

[image-1623245289393.png](#)

## Tabel splitsen (1:N)

Splits een tabel in twee tabellen en verbind de tabellen met door met de foreign key (FK) te verwijzen naar de primary key (PK) van de andere tabel.

image.1623245367763.png

## Tabel splitsen met koppeltabel (N:M)

Als je een tabel splitst en de relatie is een many to many relatie (N:M) dan heb je een koppeltabel nodig om de twee tabellen te verbinden. In de koppel tabel zitten de FK's die verwijzen naar de PK's van de twee tabellen.

image.1623245421315.png

## Opgave 1

Lees de tekst door en beantwoord de volgende vragen.

1. Hoeveel normalisatieregels zijn er?
2. Hoeveel normalisatie technieken zijn er?
3. Waarvoor dient een koppeltabel?
4. Tabellen zijn aan elkaar gekoppeld, wat/hoe zorg je voor deze koppeling?
5. Zet je voor- en achternaam in één kolom, in twee kolommen of in twee tabellen?
6. Met welk programma maak jij in jouw examen een ERD?
7. Een tabel heet "ExamenUitslagen", is dit een juiste naam? Weet jij een betere naam?
8. Waarom een goed database-ontwerp belangrijk?

## Opgave 2

1. Hieronder zie je een tabel met gegevens over gespeelde voetbalwedstrijden.

Op elke regel staat het thuis-land en gast-land.

De score van beide landen staat in de volgende twee kolommen.

De winnaar is 0,1 of 2. Bij 0 is er geen winnaar, bij 1 is het thuis-land winnaar en bij 2 is het gast-land winnaar.

De laatste twee kolommen laten het aantal kaarten (rode- en gele-) en het aantal penalties zien.

Deze opgave staat in [de Excel sheet](#).



- Aan welke normalisatieregels voldoet deze tabel *niet*?
- Verander de bovenstaande tabel zodat deze wel aan alle normalisatie-regels voldoet.

## Opgave 3

Kopieer de onderstaande tabel naar Excel (Copy/Paste, of gebruik [de Excel Sheet](#)) en normaliseer de gegevens. In de tabel staan studenten die één of meer cursussen volgens. Iedere student heeft een coach en de coach is bereikbaar via het telefoonnummer.

Gebruik de drie normalisatie technieken (kolommen en tabellen splitsen) zodat je aan de zes normalisatieregels voldoet.

- Splits de gegevens in aparte kolommen en/of tabellen zodat je aan alle regels voldoet.

Student	Cursus	Coach	telefoon_coach
Kevin Drum	Digitale Vaardigheden, Rekenen, PHP	Ayoub	06 1221 3268
Murvin Drake	PHP, JavaScript	Ayoub	06 1221 3268
John Jones, 1234	C++	Ayoub	06 1221 3268
Sally-Jane Jones	HTML, Python	Ayoub	06 1221 3268
David (getrouwd)	Rekenen, PHP	Ayoub	06 1221 3268
Murvin Drake	PHP, JavaScript	Samina	06 2100 3485
Murvin Drake	C++	Samina	06 2100 3485

## Opgave 4

In een bibliotheek wil men bijhouden welke klanten welk boek van welke periode tot periode hebben geleend.

Elke klant kan meerdere boeken gelijktijdig lenen. De bibliotheek heeft soms meerdere exemplaren van één title. Verder wil men de klant een whatsapp kunnen sturen twee dagen voor het verstrijken van de inleverdatum.

- Maak een databaseontwerp.

Begin met de onderstaande gegevens (copy paste naar Excel of gebruik [de Excel sheet](#)).

klant	boek_id	boek ISBN	Titel	uitgeleend	uitleentermijn	ingeleverd
Caris Villa	234189	817525766-0	De boze wolff	09/06/2021	21	
Caris Villa	234288	817333566-1	De lastige wereld	07/05/2021	21	14/05/2021
Caris Villa	134585	817525766-0	Op reis met JJ	28/04/2021	21	29/04/2021
Caris Villa	232269	817528866-3	Het grote feest	04/05/2021	21	25/05/2021
Ayana Tucker	232270	817528866-4	Het grote feest	29/05/2021	35	07/06/2021
Shola Greig	134585	817525766-0	Op reis met JJ	28/05/2021	21	

- Zet je databaseontwerp in [Lucichart](#).

## Opgave 5a

Een recycle bedrijf wil bijhouden uit welke verkoopbare onderdelen een apparaat bestaat. Zo bestaat een mobieltje uit een lithium-ion batterij, scherm en moederbord. Een draadloze koptelefoon bestaat uit een lithium-ion batterij en magneten.

Van elk apparaat wil men een inkoop prijs vaststellen en van alle onderdelen wil men een verkoopprijs bepalen.

[image-1623916442206.png](#)

- Maak een database in [Lucichart](#)

## Opgave 5b

De inkoop prijs van elk apparaat is een richtprijs. Als van een telefoon het scherm bijvoorbeeld is gebarsten dan is deze minder waard.

Het recycle bedrijf heeft een aantal inkopers. Elke inkoper bepaald de werkelijke prijs voor een ingekocht product.

Je wilt dus deze data ook vastleggen in de database van opgave 3a.

Inkoper	Datum	Apparaat	Standaard	Inkoop
Jan	10/May	Espresso apparaat	25,00	27,00
Jan	12/May	koptelefoon	6,00	7,00

Jan	12/May	koptelefoon	6,00	5,00
Jan	12/May	iPhone 6	50,00	45,00
Mimi	12/May	iPhone 6	50,00	40,00
Mimi	13/May	Laptop Lenovo G7800	80,00	85,00
Mimi	13/May	Espresso apparaat	25,00	30,00

Zorg ervoor dat deze data in de juiste tabellen kan worden opgeslagen. Maak het **volledige** datamodel dat aan alle normalisatieregels voldoet.

De tabel staat ook in [de Excel sheet](#).

## Voorbeeld 6

Bij autohandel "Krakkemik" staan er auto's van verschillende merken op het terrein. In hun database staat bijvoorbeeld dat er; een rode Opel Astra uit 1998 voor 1500 euro te koop is, een groene Ford Escort uit 2002 voor 2000 euro en een Peugeot waarvan de prijs 4000 euro is.

Herken jij de entiteit? Van wat wordt hier gegevens vastgelegd?

En wat zijn de attributen van de entiteit? Benoem alle attributen.

Maak het ERD in Lucichart.

## opgave 7, ToDo App

Maak een datamodel (ERD) voor een ToDo applicatie. In de ToDo applicatie wil je de naam van een ToDo item kunnen vastleggen. Van elk ToDo item wil je de titel kunnen vastleggen, bijvoorbeeld "paspoort verlengen". Verder wil je van elk ToDo item vastleggen wanneer hij is aangemaakt en of hij afgerond is.

Verder wil je de ToDo items opdelen in categorieën. Bijvoorbeeld, de categorie "boodschappen", "privé" of "werk". Van elke categorie wil je vastleggen wanneer deze voor het laatst gewijzigd is.

Maak het ERD in [Lucichart](#).

## opgave 8, Restaurant en Corona

In verband met Corona wil het restaurant de "Pittige Stake", een applicatie laten ontwikkelen waarin ze kunnen vastleggen welke klanten bij elkaar aan de tafel hebben gezeten. Op die manier

kunnen ze als iemand positief op Corona is getest, bepalen bij wie deze persoon aan tafel heeft gezeten en kunnen deze tafelgenoten worden gevraagd om zich ook te laten testen.

Van elke klant dient de voor- en achternaam en zijn telefoonnummer te worden vastgelegd. Verder moet er worden vastgelegd aan welke tafel de persoon zat en wanneer dit was (datum en tijd). De tafels in het restaurant zijn genummerd van 1 t/m 6 en alleen gasten aan dezelfde tafel dienen te worden gewaarschuwd als een tafelgenoot positief wordt getest op Corona.

Maak het ERD in [Lucichart](#).

## opgave 9, Covid-19 test

Het RIVM wil een eenvoudige database waarin ze kunnen zien welke persoon wanneer is getest en wat de uitslag was (positief of negatief). Van alle geteste hoeft alleen de voor- en achternaam en het e-mailadres te worden vastgelegd.

Het RIVM wil ook graag weten of de geteste persoon de Corona app heeft geactiveerd. Als dat zo is dan kunnen ze namelijk via de app andere mensen op de hoogte stellen als zij langere tijd bij een positief getest persoon in de buurt zijn geweest.

Maak het ERD in [Lucichart](#).

## opgave 10, Fietsenmaker Snelle Jelle

Fietsenmaker Snelle Jelle wil na een reparatiebeurt zijn klanten per SMS of Whatsapp op de hoogte stellen dat de reparatie klaar is. In dit bericht wil hij ook vertellen hoe hoog de reparatiekosten zijn.

Omdat de veel klanten meer dan één fiets hebben, wil hij van de fietsen ook wat kenmerken vastleggen. Hij wil het merk, model, type en kleur kunnen vastleggen.

Van elke reparatiebeurt wil hij verder vastleggen wanneer het onderhoud plaatsvond, hoe lang de reparatie duurde, wat er is uitgevoerd en de prijs.

Maak het ERD in [Lucichart](#).

## opgave 11, Eagle Dev \*\*

Maak het gedeelte van het datamodel van EagleDev waarin van de studenten wordt vastgelegd welke module ze hebben uitgevoerd. Van de studenten wil je de voor- en achternaam, hun email

en studentennummer vastleggen en van de modules wil je de naam, de omschrijving, de status en het niveau vastleggen. De status is *open*,  *bezig* of *klaar*. En het niveau is opgedeeld in 12 levels, 1 tot en met 12.

Maak het ERD in [Lucichart](#).

## opgave 12, garagebedrijf

image 1623242632313.png

Garagebedrijf Sneller B.V. wil graag automatisch een orderbon kunnen afdrukken. Op de orderbon moeten de klantgegevens staan en er moet op staan welke artikelen de klant heeft besteld.

Verder wil het garage bedrijf ook vastleggen welke type auto de klant heeft, wat de uitvoering is, wat het kenteken is, wat het bouwjaar is en wat de kleur is. Een klant kan meer dan één auto bezitten.

Maak een database ontwerp (ERD) in [Lucichart](#).

--



# ExamenExec (in Yii2)

## *Student*

### **Student Login**

De student gebruikt zijn studentnummer om in te loggen. Als een student eenmaal is ingelogd hoeft hij dat niet opnieuw te doen. Zijn gevalideerd studentnummer wordt in de browser opgeslagen.

### **Studentenscherm**

Het scherm voor een student heeft drie delen: (1) een overzicht van aangevraagde gesprekken, (2) een indicator van de drukte en wachttijd, en (3) een deel waarin hij een nieuw gesprek kan aanvragen. Het scherm wordt automatisch ongeveer elke minuut ververs.

## *Login*

### **Admin Login**

Via het Login-menu rechtsboven kan de admin inloggen, er is één admin account. Een admin kan meer dan één keer aanloggen.

### **Rolspeler Login**

Via het Login-menu rechtsboven kan de Rolspeler inloggen, er is één rolspeler account. Een rolspeler kan meer dan één keer aanloggen. Als de rolspeler is aangelogd dan identificeert de rolspeler zich verder met een voor hem unieke code. Deze code kan de admin opzoeken, instellen en aanpassen.

### **Security - ip-nummer**

De applicatie is afgeschermd op ip-nummer, m.a.w. jouw ip nummer moet in een whitelist staan om toegang te krijgen tot het systeem dat geldt voor alle rollen; student, rolspeler, admin.

## Rolspeler

Nadat de rolspeler met is ingelogd op een algemeen rolspeler-account wordt gevraagd een rolspelerID in te voeren. Het rolspelerID kan door de admin worden beheerd (onder Token).

Nadat de rolspeler is aangelogd, krijgt hij een overzicht van gesprekken. Vanuit dit scherm kunnen gesprekken gestart worden. Een gesprek heeft drie statussen: (1) klaar; er staat een vinkje achter het gesprek, (2) het gesprek loopt en is dus gestart maar nog niet afgerond; er staat 'herstart' achter het gesprek, en (3) open, het gesprek dient gestart te worden; er staat 'Start Gesprek' achter het gesprek.

Als een gesprek gestart wordt dan staat er per gesprekstype een korte uitleg en worden er vragen gesteld die met Ja/Nee en in bepaalde gevallen met Soms/Beetje kunne worden beantwoord. Aan het eind kan nog een opmerking worden ingevoerd. Het gesprek is definitief als er op Save wordt gedrukt.

# Admin

De admin kan alles en kan alle schermen van de studenten en de rolspelers zien.

## Admin - menu

Onder formulieren worden beoordelingsformulieren gedefinieerd.

Let op: formulier = gesprek (althans voor nu)

Formulieren hebben een nummer. Het nummer bepaald de volgorde van de formulieren in de verschillende menu's.

Formulieren hebben verder een titel (omschrijving) en een Instructie (voor de rolspeler). De instructie mag eenvoudige HTML tags bevatten.

Formulieren kunnen in het overzicht met het vinkje actief of inactief worden gezet. Een inactief formulier kan niet meer worden aangevraagd door een student. Op een dag worden vaak maar een beperkt aantal gesprekken gevoerd. De andere gesprekken kunne dan tijdelijk op inactief worden gezet.

Onder vragen worden de vragen bijgehouden. Vragen moeten precies onder een formulier vallen. Vragen hebben ook een volgnummer, deze bepaald de volgorde van de vragen op het beoordelingsformulier.

## Student

De admin kan een student login testen; hij doet alsof hij een student is en ziet hetzelfde scherm.

Omdat een ingelogde student 'persistent' is, kan de admin de student ook weer uitloggen.

## Rolspeler

De admin kan een rolspeleer login testen; hij doet alsof hij een rolspeleer is en ziet hetzelfde scherm.

# Status (Admin)

## Studenten

Overzicht van studenten. Hier kunnen studenten worden beheerd.

## Rolspelers

Overzicht van rolspeleers. Hier kunnen rolspeleers worden beheerd. Rolspeleers kunnen (tijdelijk) worden gedeactiveerd als zij bijvoorbeeld maar een dagdeel aanwezig zijn. Het token wordt gebruikt door de rolspeleer om zich te identificeren. Dit token dien uniek te zijn en dit wordt (nog) niet door de code gecontroleerd.

ToDo: check of ingevoerd token uniek is.

## Alle gesprekken

Dit is het belangrijkste overzicht voor de Admin. In dit scherm worden nieuwe gespreksaanvragen getoond. Van alle gesprekken wordt de status getoond.

Als alles goed gaat dan is het enige dat de Admin moet doen het toekennen van een rolspeleer. Voor een nieuwe gespreksaanvraag.

Als alles goed verloopt dan dient de status niet te worden aangepast.

## Status aanpassen - voorzichtig!

Wanneer zou de status wel moeten worden aangepast? Als om welke reden dan ook een gesprek over moet dan kan de status van *klaar* naar *wachten* worden veranderd. Er kan dan een nieuw gesprek worden ingepland. Het oude gesprek wordt verwijderd en is alleen nog via de audit log te raadplegen.

Een gesprek(saanvraag) kan ook handmatig verwijderd worden, bijvoorbeeld als een student per ongeluk 2x hetzelfde gesprek heeft aangevraagd. Klik hiervoor op het oogje aan het einde van de regel.

Als een gesprek klaar is kan op de eerste kolom op de tijd worden geclickt, het ingevulde vragenformulier wordt dan getoond.

ToDo: testen of tijden juist zijn (time zone op productie)

## Overzicht gesprekken

In dit overzicht wordt het aantal gesprekken per student getoond. Als je op de studentnaam klikt dan wordt het detailoverzicht van de student getoond. Dit is hetzelfde overzicht dat de student zelf ziet.

## ToDo (technical)

- Clean object en views *beoordelen*, this is still part of code for debug reasons.
- Check other objects/views for unused code.
- Change dropdown in gesprek overzicht or make it only editable in 'debug' mode?
- Create new form (formulieren) returns in (standard) update form, should go back to /form/index
- Update rolspeler returns in (standard) update form, should go back to /examen/rolspeler
- Delete rolspeler asks for "are you sure to delete this item?" -> message should contain rolspeler name.
- Error when retrieving a gesprek that was replaced by a new gesprek.

## Snapshot

(nieuwe stijl)

uitslag/create-snapshot?snapshot=01

--

# Challenges

## Code Snippets

Maak een app waarin je code snippets bewaard. Elke code snippet heeft een korte beschrijving, een onderwerp en een URL. Bijvoorbeeld:

onderwerp	PDO
beschrijving	select query
url:	<a href="https://www.w3schools.com/php/php_mysql_select.asp">https://www.w3schools.com/php/php_mysql_select.asp</a>

### Snippet:

```
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}

$conn->close();
```

### Eisen

- lijst met beschrijvingen per onderwerp
- kunnen zoeken binnen beschrijving (over alle onderwerpen over één onderwerp).

### Toepassing

Opbouwen eigen library met code snippets ter voorbereiding op je examen.

# URL Library

Vastleggen shortcuts met omschrijving en indelen per categorie.

# Rolspeler (Kerntaak-1)

## Wat doet een rolspeler?

Tijdens de praktijkexamens, Kerntaak-1 moeten de studenten in een gesimuleerde omgeving een opdracht van een klant aannemen. Tijdens dit examen (kerntaak-1) moeten de studenten alle stappen doorlopen voordat ze aan het programmeren gaan. De stappen die zij moeten uitvoeren zijn de volgende:

1. opstellen **Programma van Eisen**

Aan de hand van informatie uit onder meer een gesprek met de rolspeler moet de student een de opdracht in het kort terugkoppelen aan de klant. Het resultaat moet worden overgedragen aan de klant.

2. opstellen **Projectplan**

Na de goedkeuring van het programma van eisen moet de student een projectplan maken. Het projectplan dient aan de kant worden teriggekoppeld.

3. opstellen **Functioneel Ontwerp**

Na de goedkeuring van het projectplan moet de student een functioneel ontwerp maken en dit overdragen aan de klant.

4. opstellen **Technisch Ontwerp**

Na de goedkeuring van het functioneel ontwerp moet de student een technisch ontwerp maken en dit overdragen aan zijn projectleider.

5. klaarmaken **ontwikkelomgeving**

De ontwikkelomgeving moet worden klaargezet worden getest en worden gedocumenteerd.

Er zijn per student 5 gesprekken:

1. Aan het begin van het examen is er een gesprek tussen de klant en de student. De student moet de klant vragen wat hij precies wil laten bouwen. De rolspeler beoordeelt de student op het gesprek. Dit gesprek duur maximaal 10 minuten.
2. De student presenteert zijn Programma van Eisen en vraagt akkoord om door te gaan. De klant zal altijd akkoord gaan, want ook hier wordt het gesprek beoordeeld en niet de inhoud (van het document). Dit gesprek duurt maximaal 5 minuten.
3. De student stemt het projectplan af met de klant en vraagt om op toestemming om door te gaan. Dit gesprek duurt maximaal 5 minuten.

4. De student draagt het functioneel ontwerp over en geeft een toelichting. Dit gesprek duurt maximaal 5 minuten.
5. De student draagt het Technisch Ontwerp over en geeft een toelichting. De overdracht is aan de projectleider. Dit zal een wat technischer gesprek zijn maar ook hier gaat het niet om de inhoud, maar om het beoordelen van het gesprek. Dit gesprek duurt maximaal 5 minuten.

De rolspeler speelt de rol van de klant en/of de rol van de projectleider. De rolspeler beoordeelt de student aan de hand van een beoordelingsformulier. De beoordeling is niet inhoudelijk. Het gaat dus niet om de inhoud van het document, deze wordt door de examinatoren beoordeeld, maar het gaat om de vorm en inhoud van het gesprek.

Als rolspeler ben je (in bijna alle gevallen) een klant. Je kunt je ook als klant gedragen. Je weet weinig van IT en wilt gewoon zo snel en goedkoop mogelijk een oplossing voor jouw probleem. Probeer een zo natuurlijk mogelijk gesprek te voeren.

De meeste studenten zijn nerveus en vinden het lastig om een gesprek te voeren. Als klant mag je hier natuurlijk begrip voor tonen.

## Hoe worden de gesprekken gevoerd?

De gesprekken vinden online plaats via Teams. Via Teams wordt er een video-verbinding gemaakt door de rolspeler met de student. De gesprekken dienen via Teams te worden opgenomen.

Via het opgegeven email adres krijg je een uitnodiging om je te registreren op (Microsoft) Teams. Je krijgt een zogenaamd Guest account waarmee je gesprekken kunt voeren met de studenten of met de examinatoren.

## Teams

Via een email heb je een uitnodiging gehad om je te registreren via Teams. Je klikt op de link en je kunt een (guest) account aanmaken op de Teams site van het ROC van Amsterdam. Je kunt een aparte Teams applicatie downloaden maar je kunt Teams ook in de Browser draaien (Microsoft Edge of Chrome anders zijn bepaalde functie die je nodig hebt niet beschikbaar).

Als je al teams hebt dan kun je in jouw Teams het ROC van Amsterdam account toevoegen. Je kunt dan omschakelen van het ene account naar het andere. Beide accounts gelijktijdig gebruiken gaat heel lastig, maar je kunt dus wel redelijk snel omschakelen zonder dat je elke keer opnieuw hoeft aan te loggen.

## Beoordeling



Via de website [examen.student.ovh](https://examen.student.ovh) kunne de studenten worden beoordeeld. Om toegang te krijgen voor deze site dient jouw IP nummer worden geregistreerd. Het IP nummer van jouw computer is afhankelijk van de locatie waar vandaan je de studenten gaat beoordelen. Het IP adres niet geheim, het is adres van jouw netwerk dat bij elk verzoek door jouw browser wordt meegestuurd. Dit zorgt ervoor dat de webserver weet naar wie er een antwoord moet worden verstuurd.

Je IP-nummer vind je door bijvoorbeeld naar [deze](https://www.watismijnipadres.nl/) (<https://www.watismijnipadres.nl/>) site te gaan. Geef dit nummer van te voren door.

Verder krijg je een gebruikersnaam en wachtwoord en een zogenaamde token. Deze heb je nodig om als rolspeler te kunnen aanloggen.

Ga naar [examen.student.ovh](https://examen.student.ovh) en druk rechtsboven op login. Log in met de gebruikersnaam en wachtwoord. Op het volgende scherm vul je jouw rolspeler token (ID) in.

[image-1604781617262.png](#)

Het volgende scherm wordt getoond.

[image-1604781710929.png](#)

In het begin zal het scherm leeg zijn. Op het moment dat er een gesprek in de wacht staat, staat er 'Start Gesprek'.

Het gesprek wordt zo snel mogelijk gestart door te klikken op 'Start Gesprek'. In dit geval gaat het om een kort gesprek:

[image-1604782485966.png](#)

De student laat zijn projectplan zien via Teams en geeft een korte toelichting. In dit korte gesprek dient er slechts op twee punten te worden beoordeeld.

[In Teams staat een overzicht van alle vragenlijsten.](#)

Mocht het er iets gebeuren, dan kan er opnieuw worden aangemeld en kan het gesprek worden hervat. nadat er op 'Save' wordt gedruk is de beoordeling definitief.

In de opmerking kunt je extra observaties delen. Bijvoorbeeld; "het projectplan werd pas over gedragen nadat ik er na vroeg" of "student was erg nerveus en kwam slecht over" of "er is iet nadrukkelijk om goedkeuring gevraagd maar de student vroeg of het zo goed was", of "prima gesprek, student kwam goed over".

Let op de maximale gesprekstijd en de instructie die bovenaan op de pagina staat.

## Procedure

## Algemene kanaal

Via Teams kan er worden gecommuniceerd. Dat kan in het algemene kanaal. Het [algemene kanaal](#) kan iedereen die betrokken is bij het examen zien, rolspelers, docenten en studenten.

## Privé boodschap

En dat kan ook via een privé boodschap. Je kunt mij een [privé boodschap sturen door hier op te klikken.](#)

Als rolspeler ben je beschikbaar tijdens de afgesproken tijd, mocht je even willen pauzeren, geef dat dan aan via Teams. Je wordt dan niet ingedeeld totdat je weer meldt dat je aanwezig bent.

Studenten lopen met hun laptop naar een 'vergader-locatie', als ze daar zitten dan zal de gespreksaanvraag in jouw scherm zichtbaar zijn. Als er een gesprek klaar staat dan zit een student dus te wachten om gebeld te worden. Het verzoek is dan om de betreffende student zo snel mogelijk te (beeld)bellen via Teams.

## Bellen

Via de chat functie type je de naam van de student in en daarna druk op je het video icoontje recht boven in het scherm.

[image-1604875423258.png](#)

In het antwoordformulier kan ook op de naam worden geklikt. Als Teams is geïnstalleerd dan moet je die wellicht één keer kopellen maar daarna kun je door op de naam te klikken direct naar de juiste persoon in Teams gaan.

[image-1604996699534.png](#)

## Geen beeld?

De student heeft beeld en geluid aan. Mochten er om welke reden dan ook geen beeldverbinding mogelijk zijn, bijvoorbeeld door een slechte WiFi verbinding, dan kan worden uitgeweken naar audio alleen. In beide gevallen moet het gesprek worden opgenomen.

## Opname

He gesprek dient te worden opgenomen. Ga hiervoor naar de ... (drie puntjes) bovenaan in het scherm en kies "Start Recording".

[image-1604917273768.png](#)

De opname wordt in de Teams opgeving bewaard en de opname stopt als het gesprek wordt beëindigd.

## Opname Guest accounts (moet nog worden getest)

Voor externe rolspelers die via een guest account toegang tot Teams hebben geldt dat zij **geen** video-opname kunnen maken. Voor hen geldt dat een audio opname volstaat. Met Windows 10 gaat dat met De 'Voice Recorder' de opnames worden geplaatst als MP3 bestand in de documents folder en dan in 'Sound Recordings'. Hernoem na elk gesprek de laatste recording. Onderaan in het beoordelingsformulier staat een naam voor het audio bestand:

[image.1604954405307.png](#)

Klik op de naam, ga naar het bestand, selecteer het bestand en met Ctrl-V geef je het bestand de juiste naam. Let op je kunt het beoordelingsformulier nadat deze is afgesloten vanuit jouw overzichtsscherm weer openen om de juiste naam te kopiëren.

Stuur het bestand via Teams op naar Max Bisschop.

## De case

Waar gaat het over, wat moet de student ontwerpen? Dit zal kort voor het examen met de rolspelers worden doorgenomen.

## De gesprekken

De meeste gesprekken zijn kort en de student moet zijn document presenteren en om goedkeuring vragen. De goedkeuring is een pro forma vraag het antwoord van de rolspeler zal altijd positief zijn.

## De antwoorden

In het eerste gesprek wordt u als klant gevraagd over wat er gebouwd moet worden. De student heeft in principe alle informatie al één keer gehoord, maar zal toch nog alle een keer met u doornemen. De student vraagt inhoudelijke vragen over de case. Alle informatie die je als klant moet/mag delen wordt van tevoren met je gedeeld. Mochten er onverwachtse vragen zijn over details dan kun je daar een antwoord op verzinnen. Voorbeeld "welke kleur moet de OK button worden?", of "hoe hoog moet de banner zijn?". Deze laatste vraag is misschien te technisch en als klant kun je natuurlijk vragen wat een banner is. In dit geval wordt waarschijnlijk het logo dat bovenaan de webpagina staat bedoeld.

## Antwoord onbekend

Mochten er vragen zijn waar je echt geen antwoord op weet, noteer dan de naam van de student en vertel hem dat hij het antwoord via Team zo snel mogelijk zal ontvangen. na het gesprek kun je de vraag via teams naar de examen coördinator doorsturen die zal voor de afhandeling zorgen.

# Test

In de week voor het examen zal de coördinator van het examen via Teams contact opnemen. We zullen dan nog een keer alle stappen doornemen en controleren of alles werkt.

## Checklist

1. Kun je in Teams?
2. Heb je jouw IP nummer doorgegeven (<https://www.watismijnipadres.nl/>)
3. Kun je aanloggen als rolspeler op examen.student.ovh ?

## Tijdsschema

Er zijn twee groepen die in de planning die een half uur van elkaar verschillen. Niet elke student heeft evenveel tijd nodig en er zijn ook studenten met studieverlenging. Het is dus moeilijk precies weer te geven wanneer welke gesprekken plaatsvinden. We hebben wel een schatting gemaakt en die loop als volgt:

### Dag 1

[image-1605030937569.png](#)

[image-1605031085782.png](#)

### Dag 2

Er vindt dinsdag slechts één gesprek plaats. Tussen 10:30 - 13:30 zullen de meeste gesprekken worden gevoerd. Na 13:30 uur hebben we slechts één of twee rolspelers nodig voor kandidaten die verlenging hebben.

--

# PHP Forms en Arrays

## Wat leer je in deze les?

1. Hoe werken forms, wat is de action van een form en hoe maak je een verbinding tussen een form en (PHP) code.
2. We leren output van een script in een HTML table te plaatsen.
3. Hoe werken arrays en hoe kan je dit toepassen.
4. Hoe werken associatieve arrays en hoe kan je dit toepassen?

We gaan telkens zelf een stap maken in duo's.

## Stap 1

Gebruik HTML en maak een form dat er zo uit ziet:

[image-1605803026327.png](#)

## Stap 2

Zorg ervoor dat als je op submit drukt je naar een PHP-pagina gaat en die het volgende afdruckt.

[image-1605803189938.png](#)

## Stap 3

Verander de output nu en druk de velden die je in het formulier ziet af.

Je vult dit in en drukt op submit en...

[image-1605803316418.png](#)

...dan zie je dit.

[image-1605803351329.png](#)

## Stap 4

Probeer nu je PHP script aan te passen zodat je in plaats van een 2 de tweede dag van de week afdruckt.

Als je 1 invult dan laat het script 'maandag' zien, als je 2 in vult 'dinsdag', als je 3 invult 'woensdag', enzovoorts. De output ziet er dus zo uit:

image-1605804657444.png

Tip1: gebruik hiervoor een array.

## Stap 5

Kun jij de output ook netjes in een (html) array zetten? De output ziet er dan ongeveer zo uit:

image-1605804736956.png

Willen we het nog fancyer dan kunnen we heel eenvoudig Bootstrap gebruiken.

image-1605805171603.png

## Stap 6

Hoe heb je stap 4 opgelost? 'dinsdag' is het 2de element uit het array, maar de array index waarmee je de waarde laat zien is 1.

Je zou dit ook kunnen oplossen met een associatief array. In een 'gewoon' array loopt de index van 0 en telt die gewoon door, dus 0,1,2,3,4,..... bij een associatief array bepaal jezelf de index.

Bijvoorbeeld:

```
$array=['Mike'=>8, 'Ayoub'=>6, 'Niels'=>3, 'Adam'=>5 ];  
echo $array['Niels'];
```

Wat wordt er afgedrukt?

Kun je in de vorige stap ook gebruik maken van een associatief array? Zo ja hoe dan?

## Stap 7 - de uitdaging!

Nu gaan we een hele lastige constructie opzetten. We gaan een associatief array opzetten en de value/waarde van elke element is op zijn beurt weer een 'gewoon' array.

In plaats van de leerlingen nu allemaal één cijfer te geven, geven we ze ieder 3 cijfers.

```
$array=['Mike'=>[8,8,5] 'Ayoub'=>[5,6,7] 'Niels'=>[3, 10, 8] 'Adam'=>[5, 9, 7];
```

Maak nu een PHP-programmaatje om *alle* cijfers van Niels af te drukken.

--



# Intake

[image-1610976349431.png](#)

---

## Beste Student,

Jullie gaan een intake opdracht maken. Volg de stappen hieronder:

1. Download een editor, bijvoorbeeld [Brackets](#)
2. Download het [bestandje](#) in de bijlage.
3. Start de editor (uit stap 1) en laadt het bestandje uit stap 2..
4. Download de [instructie](#) en voer deze stap-voor-stap-uit.



# Framework

## Les 1, opdracht 1

1. Bespreek met elkaar wat een framework is en probeer een omschrijving te maken.
  2. Bespreek met elkaar waarom jij een framework zou willen leren gebruiken?
  3. Welke (PHP) frameworks ken je?
- 

## Les 1, opdracht 2

**Als jij een framework kiest om te leren, wat is dan belangrijk voor jou? Zet de volgende punten op volgorde van belangrijkheid (van meest belangrijk naar minder).**

- Veel over te vinden (tutorials e.d.).
- Veel plugins voor beschikbaar.
- Wordt actief onderhouden, veel updates dus.
- Is makkelijk te leren, kun je makkelijk je examen mee halen.
- Kun je snel iets mee maken/ontwikkelen.
- Heeft heel veel mogelijkheden.
- Genereert automatisch code, die je dan kan aanpassen.
- Heeft veel 'straight out of the box', sorteren, selecteren, login, menu, etc.
- Is hip/modern/cool.

# ExamBot

Een examan

# Kerntaak 3

## Overzicht KT3

[!\[\]\(c507f772dba2b921f86777f01218e570\_img.jpg\)](#)

[!\[\]\(4729e517bc6a7cd81c8025b9646574fb\_img.jpg\)](#)

## Opdracht 1

Download de template en stel van alle (vijf) de functionaliteiten (in grijs) minimaal één testscenario op.

De case wordt hier beschreven:

<https://www.roc.ovh/books/examentraining-b1-k1/page/case-project-kinderopvang>

De template staat hier:

[Oefenopdracht testscenario \(KT3\).docx](#)

## Opdracht 2

Controleer of je een demo filmpje kan maken, de instructie staat hier:

[Filmpje voor examen maken](#)

## Opdracht 3

Dit onderdeel kan alleen worden getest als je een login op de productieserver heb gekregen.

Controleer of je een web sites live kunt zetten, de instructie staat hier:

[Naar productie](#)



# KT Beoordeling

## Ga naar Teams en kopieer 4 bestanden



Elk bestand hoort bij een kerntaak.

Per Excel file zijn meerdere tabjes te vinden. Elk tabje hoort bij een werkproces.



Open Padlet: <https://padlet.com/maxbisschop1/mfth0dsmj6dl6wc7>

Maak nu gezamenlijk ene lijst met zaken die volgens de beoordelingscriteria belangrijks zijn.

## Codeerstijl

Nette code is belangrijk voor het examen.

Hoe ging het ook alweer met inspringen van code: <https://youtu.be/2ja9gu3ANME>

--

# MariaDB

## Change timestamp on update and insert

```
ALTER TABLE <table name>  
CHANGE  
  `<column name>` datetime  
TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
```

# ED Status 3de jaars

<a href="#">Repo's</a>	<a href="#">MySQL Adv</a>	<a href="#">PHP exp1</a>	<a href="#">PHP exp2</a>	<a href="#">PHP adv.</a>	<a href="#">MySQL basic</a>	<a href="#">PHP Basic</a>	<a href="#">DB Design</a>
Mohamed							
Colin							
Tarik						V	
Faissal					V		
Tyrone						V	
N'Joy					V	V	
Lucien				V		V	
Dardan	V			V	V	V	
Styn	V			V	V	V	
Shano	V			V	V	V	V
Jochem	V			V	V	V	V
Donne	V	V		V	V	V	
Elyesa	V	V	V	V	V	V	

# Checklist DB Design (EN)

## The 5 Basic Rules

1. An **entity** is a person, thing or event. A number (for example weight) is never an entity but it is an attribute (characteristic) of an entity.
2. Every entity has exactly one **PK (primary key)**. The primary key is unique for the entity. For example social security number for a person or licence plate number for a car.
3. Entities can have the following **relations**: 1:1, 1:N, N:1 or N:M.
  - 1:1 relations are rare. If they occur, you can most probably merge the two relations into one.
  - 1:N and N:1 are the same and will occur in most cases.
  - N:M relation is translated into two 1:N relations via a so called connection-entity/table.
4. A **1:N** relation is drawn via a line. The **line** has a **triangle** or rake on one side and just a line on the other side. The line is connected to the one-side and the triangle connects the many-side.
5. Every triangle (rake) 'belongs' to one FK. The FK is connected to the PK of the connecting entity and has the same data type (the name of the attribute, although the same in content, may differ).

## Data types

The most common data-type are:

Datatype	Voorbeeld
int	-2 147 648 - 2 147 649
varchar(), bijv. varchar(20)	"Big Boss 12"




date	2022-04-01
datetime	2022-04-01 18:43:12
decimal(6,2)	1250,95

## Relation(1:N) between entities

Een relatie tussen twee entiteiten is vrijwel altijd een 1-op-meer relatie en heeft daardoor aan één kant een 'harkje'.

A relation of two entities is almost always an 1 to many relation. The many side of the relation had the triangle or rake ('harkje' in Dutch).

 ('harkje')

### Examples

- One person owns more (0, 1 or more) cars and one car belongs to exactly one person.
- One school class consists out of more (most commonly more than 1) students. One student belongs to one class.
- One home work assignment can be submitted 0,1 or more times (think about Canvas). One assignment submission belongs to one student.
- One football team has more players and one player belongs to one football team.
- One school has more students and one student is registered to one school.

Note that all these cases are describing the most common situation. Of course, you could think of one student who is registered at two schools, but that would be a rare case. If this still would not be a very rare case and you want to design your database so that one student can be registered at 2 or more schools, you would end up with an N:M relation which is more complex and will be described below.

## Example

Suppose you have two entities, **student** and **study coach**. In order to determine the relation, ask yourself what is applicable:

one study coach	more students
-----------------	---------------

one student	more study coaches
-------------	--------------------

Both are possible in theory, but in the situation at our school only the first line applies.

This means that the more side will be the student, hence it will get the triangle. The relation between the two relations has the triangle on the student side and a plain line on the study coach side. The student will get a FK which will connect to the PK of the study coach.

	one side	more side
PK	has one unique PK	has one unique PK
Lijntje	No triangle just a line	Triangle ('harkje')
FK	has no FK	one FK which points to the PK of the one side

So the ERD will look like:

[image 1643791334313.png](#)

Both entities have gotten a unique PK. The FK and triangle are on the same side.

Summarized: the triangle is situated at the many side and with every triangle comes a FK.

Or in short:

**Triangle = More = FK**

## N:M (veel-op-veel)

Suppose you have two entities *product* and *client* (Dutch "*klant*").

The relation between these entities is N:M, many to many. One client can buy more products and one product can be bought by many clients.

In order to put this kind of relation into an (relational) database, you have to create a connecting entity/table. The ERD will look like this:

[image 1664052809552.png](#)

(Note that the attribute data types are omitted in this ERD)

The entity `product_klant` (product-client) is the connecting entity. This entity connects the product entity and client (klant) entity in such a way that the relation between client and product is a N:M relation.

De entiteit `product_klant` is de koppeltabel en deze verbind het product en de klant aan elkaar zodat er een N:M relatie ontstaat.

Note that the connecting table has two FK's (it also has two triangles). The combinations of these FK's are connecting a *product* to a *client*.

Suppose you have two clients, one with id (PK) 101 and one with id 102. Both clients bought product with the id 10 and 11. The content of the connecting table will look like this:

id (PK)	klant_id (FK)	product_id (FK)
1	101	10
2	102	11
3	101	10
4	102	11

## Creating an ERD step by step

All steps to create an ERD are listed below.

1. Determine the entities. These are persons, things, or events about which you want to store information in the database.
2. Determine from every entity what attributes (data items) you want to store in the database. Every attribute has a unique name in the entity (f.e. you can't have two "id" or two "names" in the same entity).
3. Determine the right data types of every attribute.
4. Create precisely one PK in every entity. When in doubt, you can always create an (unique) id.
5. Determine the relations between the entities. For an N:M relation, insert a connecting table.
6. Draw the relations with the triangle on the more side.

7. With every triangle comes a foreign key. This FK points to the PK of the entity it connects to via the relation.
8. The PK and FK connecting to each other don't necessarily need to have the same name, but they need to have the same data type.
9. Re-read the case and check if all data that needs to be stored is somehow part of your ERD/

## What often goes wrong?

1. Does all entities has exactly one PK (*PrimaryKey*)?
2. Are all PK's unique and can only occur once in the data set?  
For example, surname cannot be PK, because there are people with the same surname.  
Tip, you can always use (your own) id as PK.
3. Does every attribute has a datatype?
4. Phone number is not an int because the first 0 will disappear, nor can you use spaces.  
Since you don't need to make any calculations with phone numbers, there is no need to make it an integer.
5. For any date use *date* as datatype.
6. If you need to record the time next to the date, use *datetime*.
7. Every relation, line can have only one triangle. This triangle is situated on the more side of the relation.
8. For every triangle you need one FK. This FK connects the entity to the related entity via it's PK.
9. The PK and FK which are related have the same data type.
10. The length of an int doesn't matter. Nowadays, all int's will use the same length. You may omit the length.
11. Varchar always has a length. This length is the maximum length that can be used for this attribute. Using varchar(5) for the attribute surname is wrong because obviously it is too short. Varchar(300) for a phone number is wrong because it is too big and will waste database space.
12. String data types do not exist. Use *char()* or *varchar()* Char has a fixed length and varchar has a maximum length.



# Windows Firewall

Als je XAMPP hebt draaien dan heb je een webserver draaien die je met *localhost* via jouw browser kan bereiken.

Je zou XAMPP ook via het locale netwerk of zelfs via internet kunnen bereiken. Dat wil je natuurlijk niet!

Standaard staat Windows zo ingesteld dat het *geen* verkeer binnen laat. Daar zorgt de firewall voor.

Hoe zet je jouw firewall aan, of zorg je dat deze aan staat?

## Windows Firewall aanzetten

1. Klik op het vregrootglas in jouw taakbalk en type **firewall** in, je ziet dit:

image1649094071710.png

2. Start de **Windows Defender Firewall**
3. Klik op Turn Windows Defender Firewall on or off.

image1649094230368.png

4. Controleer of de instellingen zo staan als hieronder staat weergegeven:

image1649094284503.png

5. Staan het niet zo, selecteer dan twee maal Turn on Windows Defender Firewall en klik onder aan de pagina op OK.

## Testen = zeker weten!

Stel je wilt echt zeker weten of jouw XAMPP inderdaad niet te bereiken is dan kun je dat als volgt testen

1. Zorg ervoor dat jouw telefoon (of ander apparaat met een browser) en jouw laptop op hetzelfde WiFi netwerk zitten.  
Gerbruik je jouw telefoon? Zorg er dan voor dat deze via WiFi op jouw netwerk zit, zet

eventueel 4G of 5G uit.

2. Ga naar windows en open een CMD windows, type in **ipconfig**. Je ziet iets dat lijkt op het volgende:

[image-1649094708021.png](#)

3. Onthoud het IPv4 Address, ga naar jouw browser op je telefoon en type dit adres in.

**Bijvoorbeeld:** <http://192.168.111.123>

4. Zie je niets dan is het goed! Je kunt eventueel *heel even* jouw firewall uit zetten en controleren of je dan wel jouw webserver ziet.

**Vergeet je firewall niet weer aan te zetten!**

## Help mijn game of ander programma werkt niet!

Sommige programma's willen met jouw laptop praten. Dit is altijd verdacht en in principe niet nodig!

Als je jouw firewall uit zet dan kan een andere computer zonder dat jij dat weet of zonder dat jij iets doet contact maken met jouw laptop.

**Zet nooit jouw firewall uit!**

Wil je om één of andere manier toch jouw firewall uit zetten, dan kun je in plaats van jouw hele firewall uit zetten ook een klein 'gaatje' in jouw firewall maken. Het gaat nu hier te ver om dat uit te leggen en ook dit is niet zonder risico.

Er geldt: pas jouw firewall settings niet aan als je niet heel goed weet wat je aan het doen bent. Voor dat je het weet wordt jouw laptop ingezet voor een DDoS attack, wordt jouw laptop misbruikt voor het minen van crypto valuta of worden jouw privé gegevens openbaar gemaakt.

## Router = firewall?

In principe is de (WiFi)router ook een firewall, mits die goed is ingesteld. Dat betekent dat je voor aanvallen van buiten jouw lokale netwerk een extra beveiligingslaag hebt. Deze laag is echter niet altijd voldoende en realiseer je dat je op school op een heel groot lokaal netwerk zit. Iedereen die op dit netwerk zit, kan jouw laptop benaderen als je geen Windows Firewall hebt aangezet.

# Meer info?

<https://laatjeniethackmaken.nl/>

<https://nl.wikihow.com/Voorkomen-dat-je-gehackt-wordt>

--



# PDG - cheat sheet

## *De vier leerstijlen van Kolb*

Voor een goed leerproces doorloop je alle fasen (in willekeurige volgorde).

Fase	Waar staat deze fase voor?	Leerstijl	
1	Concreet ervaren	Accomoderator/Doener	Doen - nadoen
2	Waarnemen en overdenken (reflectie)	Disvergeerder/Bezinner	Bekijken, lezen, overdenken
3	Abstracte begripsvorming	Assimilator/Denker	Verbanden ontdekken
4	Actief experimenteren	Convergeerder/Beslisser	Doen - nieuw creatief

## Model volledige instructie

1. Wat moet je doen?
2. Hoe moet je dat aanpakken?
3. Hulp, bij wie en waar?
4. Tijd, hoeveel krijg je?
5. Uitkomst, wat doen we ermee?
6. Klaar, wat ga je doen?

## Directe instructiemodel

1. Terugblik
2. Presentatie
3. Begeleid oefenen - streef naar hoge succesratio -> samen of in groepjes
4. Individuele verwerking
5. Terugblik

# Feedbackregels

1. Nu, tijdig
2. Positief in de groep, negatief één-op-één.
3. Concreet
4. Houd rekening met (verlegen) personen
5. Regelmatig
6. In proportie
7. Oprecht
8. Beschrijf impact
9. Persoonlijk (niet via Teams)
10. Beschrijf feiten, geen gevoel

# STARR

1. Situatie
2. Taak
3. Actie
4. Resultaat
5. Reflectie

# Korthagen

1. handelen
2. terugblikken
3. bewustwording
4. alternatief
5. uitproberen

# Roos van Leary

Dominantie: Boven - onder

Relatie: Samen - tegen

# Database Design (En)

*In this lesson we are going to practice to create a datamodel. When developing software a good datamodel is very important, it is the foundation of the application. Any mistake in the datamodel could lead to large issues in the application later.*

***A good applicaiton begins with a well designes datamodel!***

## Introduction

The steps you have to follow in order to create a datamodel is called normalization.

An ERD, the **Entity Relation Diagram**, is schematic representation of the datamodel.

An ERD consists of entities, attributes and relations and keys.

### Entity

These are (almost always) people, things or events you want to store information about. For exampl, students (people), sickness notification (event), an article (thing).

### Attributes

Artrributes are characteristics of entities. For example, the name of a student, the date of a sickness notification or a price of an article.

### Relaties

Relations are connections between entitiies. You can indentify relations by creating a sentence like :

**Each** student has registered 0,1 or more sickness notifications.

**Every** sickness notifactions is about exactly one student.

This example is an 1:N realtion. One student has 0,1, or more sickness notifications and every sickness notiifcation has exactly 1 student related to thsi notifaction. So we say the relation student:sickness notification = 1:N

### Keys

Keys are (special) attributes that are needed in order to create the relation.

To begin with, every entity has a primary key. The primary key uniquely identifies an instance of the entity. For example, a student number (identifies a student) or a license plate number (identifies a car).

The foreign key is an attribute in the entity that has an N:1 relation to another entity. The foreign key is a copy of the primary key it is referring to. In the above mentioned example, the sickness notification entity should get an FK with consists of the student number. This way the relation is made to exactly one student.

## How to create an ERD?

Every assignment begins with a story, we usually call this the case. Start by reading the case and determine what you want to register in the database. Try to identify the entities (people, things and events). Determine of every entity what to register, these are the attributes of the entity. Determine of every attribute the data type (integer, float, date, time, datetime or boolean). Determine the relations. Determine or create the primary keys (PK's) and insert the foreign keys (FK's).

The steps are:

1. Identify the entities,
2. Determine what you want to register of every entity, these are called the attributes.
3. Determine the data-type of every attribute.
4. Determine the relation between the entities.
5. Determine the PK's and FK's.

Datamodels can be compiled in [Lucichart](#), when the model is ready you can export a PDF file.

[image.1603478193579.png](#)

In order to create an ERD in [Lucichart](#) you have to create a free account. Once registered, search for the ERD-template "*Database ER Diagram*", see picture above.

### Naming conventions

The ERD is created in the English language. For naming entity- and attributes names are **Snake Case** (in lower case) is used, for example:

```
Entiteit: user
Attribuut 1: user_name
Attribuut 2: user_login_count
Attribuut 3: last_login_date
Attribuut 4: password
```

Also note that entity names are singular. For example, use student (*not* students).

## Example ERD

[image 1603486029048.png](#)

This diagram is made with Lucichart. Do you recognize the entities and attributes? What can you say about the relation? What information is being stored in this database? Do you recognize the foreign key and where it refers to?

# The normalisation rules

The process for creating a valid database design is called normalisation. It consists out of a few concrete steps.

When you don't see all entities right away you can use the formal normalisation steps. Put all data in one big overview, for example in Excel. Start with this one entity and divide the entity in two or more entities until these rules are satisfied:

1. Every attribute consists out of one data item.
2. Data in the entire columns has the same meaning.
3. Every data line is unique.
4. Columns cannot be repeated like (field\_1, field\_2, field\_3, etc.)
5. Try to minimize double data of strings. In order to edit data on one location.
6. Editing data can always be done by modifying only one column, so there are no dependencies between data in the different columns.

These rules can all be summarized as follows: make sure that every field on every line can be changed without the need to change other data items.

If your datamodel does not comply to all these rulesm you have to split your data into more entities. There are three ways to do this.

# De normalisatie technieken

## Kolommen splitsen

Splits een kolom in twee of meer kolommen.

[image-1623245289393.png](#)

## Tabel splitsen (1:N)

Splits een tabel in twee tabellen en verbind de tabellen met door met de foreign key (FK) te verwijzen naar de primary key (PK) van de andere tabel.

[image-1623245367763.png](#)

## Tabel splitsen met koppeltabel (N:M)

Als je een tabel splitst en de relatie is een many to many relatie (N:M) dan heb je een koppeltabel nodig om de twee tabellen te verbinden. In de koppel tabel zitten de FK's die verwijzen naar de PK's van de twee tabellen.

[image-1623245421315.png](#)

## Opgave 1

Lees de tekst door en beantwoord de volgende vragen.

1. Hoeveel normalisatieregels zijn er?
2. Hoeveel normalisatie technieken zijn er?
3. Waarvoor dient een koppeltabel?
4. Tabellen zijn aan elkaar gekoppeld, wat/hoe zorg je voor deze koppeling?
5. Zet je voor- en achternaam in één kolom, in twee kolommen of in twee tabellen?
6. Met welk programma maak jij in jouw examen een ERD?
7. Een tabel heet "ExamenUitslagen", is dit een juiste naam? Weet jij een betere naam?
8. Waarom een goed database-ontwerp belangrijk?

## Opgave 2

1. Hieronder zie je een tabel met gegevens over gespeelde voetbalwedstrijden.

Op elke regel staat het thuis-land en gast-land.

De score van beide landen staat in de volgende twee kolommen.

De winnaar is 0,1 of 2. Bij 0 is er geen winnaar, bij 1 is het thuis-land winnaar en bij 2 is het gast-land winnaar.

De laatste twee kolommen laten het aantal kaarten (rode- en gele-) en het aantal penalties zien.

Deze opgave staat in [de Excel sheet](#).



- a) Aan welke normalisatieregels voldoet deze tabel *niet*?
- b) Verander de bovenstaande tabel zodat deze wel aan alle normalisatie-regels voldoet.

## Opgave 3

Kopieer de onderstaande tabel naar Excel (Copy/Paste, of gebruik [de Excel Sheet](#)) en normaliseer de gegevens. In de tabel staan studenten die één of meer cursussen volgens. Iedere student heeft een coach en de coach is bereikbaar via het telefoonnummer.

Gebruik de drie normalisatie technieken (kolommen en tabellen splitsen) zodat je aan de zes normalisatieregels voldoet.

- Splits de gegevens in aparte kolommen en/of tabellen zodat je aan alle regels voldoet.

Student	Cursus	Coach	telefoon_coach
Kevin Drum	Digitale Vaardigheden, Rekenen, PHP	Ayoub	06 1221 3268
Murvin Drake	PHP, JavaScript	Ayoub	06 1221 3268
John Jones, 1234	C++	Ayoub	06 1221 3268
Sally-Jane Jones	HTML, Python	Ayoub	06 1221 3268
David (getrouwd)	Rekenen, PHP	Ayoub	06 1221 3268
Murvin Drake	PHP, JavaScript	Samina	06 2100 3485
Murvin Drake	C++	Samina	06 2100 3485

## Opgave 4



In een bibliotheek wil men bijhouden welke klanten welk boek van welke periode tot periode hebben geleend.

Elke klant kan meerdere boeken gelijktijdig lenen. De bibliotheek heeft soms meerdere exemplaren van één title. Verder wil men de klant een whatsapp kunnen sturen twee dagen voor het verstrijken van de inleverdatum.

- Maak een databaseontwerp.

Begin met de onderstaande gegevens (copy paste naar Excel of gebruik [de Excel sheet](#)).

klant	boek_id	boek ISBN	Titel	uitgeleend	uitleentermijn	ingeleverd
Caris Villa	234189	817525766-0	De boze wolff	09/06/2021	21	
Caris Villa	234288	817333566-1	De lastige wereld	07/05/2021	21	14/05/2021
Caris Villa	134585	817525766-0	Op reis met JJ	28/04/2021	21	29/04/2021
Caris Villa	232269	817528866-3	Het grote feest	04/05/2021	21	25/05/2021
Ayana Tucker	232270	817528866-4	Het grote feest	29/05/2021	35	07/06/2021
Shola Greig	134585	817525766-0	Op reis met JJ	28/05/2021	21	

- Zet je databaseontwerp in [Lucichart](#).

## Opgave 5a

Een recycle bedrijf wil bijhouden uit welke verkoopbare onderdelen een apparaat bestaat. Zo bestaat een mobieltje uit een lithium-ion batterij, scherm en moederbord. Een draadloze koptelefoon bestaat uit een lithium-ion batterij en magneten.

Van elk apparaat wil men een inkoop prijs vaststellen en van alle onderdelen wil men een verkoopprijs bepalen.

[image1623916442206.png](#)

- Maak een database in [Lucichart](#)

## Opgave 5b

De inkooprijs van elk apparaat is een richtprijs. Als van een telefoon het scherm bijvoorbeeld is gebarsten dan is deze minder waard.

Het recycle bedrijf heeft een aantal inkopers. Elke inkoper bepaald de werkelijke prijs voor een ingekocht product.

Je wilt dus deze data ook vastleggen in de database van opgave 3a.

Inkoper	Datum	Aparaat	Standaard	Inkoop
Jan	10/May	Espresso aparaat	25,00	27,00
Jan	12/May	koptelefoon	6,00	7,00
Jan	12/May	koptelefoon	6,00	5,00
Jan	12/May	iPhone 6	50,00	45,00
Mimi	12/May	iPhone 6	50,00	40,00
Mimi	13/May	Laptop Lenovo G7800	80,00	85,00
Mimi	13/May	Espresso aparaat	25,00	30,00

Zorg ervoor dat deze data in de juiste tabellen kan worden opgeslagen. Maak het **volledige** datamodel dat aan alle normalisatieregels voldoet.

De tabel staat ook in [de Excel sheet](#).

## Voorbeeld 6

Bij autohandel “Krakkemik” staan er auto's van verschillende merken op het terrein. In hun database staat bijvoorbeeld dat er; een rode Opel Astra uit 1998 voor 1500 euro te koop is, een groene Ford Escort uit 2002 voor 2000 euro en een Peugeot waarvan de prijs 4000 euro is.

Herken jij de entiteit? Van wat wordt hier gegevens vastgelegd?

En wat zijn de attributen van de entiteit? Benoem alle attributen.

Maak het ERD in Lucichart.

## opgave 7, ToDo App

Maak een datamodel (ERD) voor een ToDo applicatie. In de Todo applicatie wil je de naam van een ToDo item kunnen vastleggen. Van elk ToDo item wil je de titel kunnen vastleggen, bijvoorbeeld "paspoort verlengen". Verder wil je van elk ToDo item vastleggen wanneer hij is aangemaakt en of hij afgerond is.

Verder wil je de ToDo items opdelen in categorieën. Bijvoorbeeld, de categorie "boodschappen", "privé" of "werk". Van elke categorie wil je vastleggen wanneer deze voor het laatst gewijzigd is.

Maak het ERD in [Lucichart](#).

## opgave 8, Restaurant en Corona

In verband met Corona wil het restaurant de "Pittige Stake", een applicatie laten ontwikkelen waarin ze kunnen vastleggen welke klanten bij elkaar aan de tafel hebben gezeten. Op die manier kunnen ze als iemand positief op Corona is getest, bepalen bij wie deze persoon aan tafel heeft gezeten en kunnen deze tafelgenoten worden gevraagd om zich ook te laten testen.

Van elke klant dient de voor- en achternaam en zijn telefoonnummer te worden vastgelegd. Verder moet er worden vastgelegd aan welke tafel de persoon zat en wanneer dit was (datum en tijd). De tafels in het restaurant zijn genummerd van 1 t/m 6 en alleen gasten aan dezelfde tafel dienen te worden gewaarschuwd als een tafelgenoot positief wordt getest op Corona.

Maak het ERD in [Lucichart](#).

## opgave 9, Covid-19 test

Het RIVM wil een eenvoudige database waarin ze kunnen zien welke persoon wanneer is getest en wat de uitslag was (positief of negatief). Van alle geteste hoeft alleen de voor- en achternaam en het e-mailadres te worden vastgelegd.

Het RIVM wil ook graag weten of de geteste persoon de Corona app heeft geactiveerd. Als dat zo is dan kunnen ze namelijk via de app andere mensen op de hoogte stellen als zij langere tijd bij een positief getest persoon in de buurt zijn geweest.

Maak het ERD in [Lucichart](#).

## opgave 10, Fietsenmaker Snelle Jelle

Fietsenmaker Snelle Jelle wil na een reparatiebeurt zijn klanten per SMS of Whatsapp op de hoogte stellen dat de reparatie klaar is. In dit bericht wil hij ook vertellen hoe hoog de reparatiekosten zijn.

Omdat de veel klanten meer dan één fiets hebben, wil hij van de fietsen ook wat kenmerken vastleggen. Hij wil het merk, model, type en kleur kunnen vastleggen.

Van elke reparatiebeurt wil hij verder vastleggen wanneer het onderhoud plaatsvond, hoe lang de reparatie duurde, wat er is uitgevoerd en de prijs.

Maak het ERD in [Lucichart](#).

## opgave 11, Eagle Dev \*\*

Maak het gedeelte van het datamodel van EagleDev waarin van de studenten wordt vastgelegd welke module ze hebben uitgevoerd. Van de studenten wil je de voor- en achternaam, hun email en studentnummer vastleggen en van de modules wil je de naam, de omschrijving, de status en het niveau vastleggen. De status is *open*,  *bezig* of *klaar*. En het niveau is opgedeeld in 12 levels, 1 tot en met 12.

Maak het ERD in [Lucichart](#).

## opgave 12, garagebedrijf

[image1623242632313.png](#)

Garagebedrijf Sneller B.V. wil graag automatisch een orderbon kunnen afdrukken. Op de orderbon moeten de klantgegevens staan en er moet op staan welke artikelen de klant heeft besteld.

Verder wil het garage bedrijf ook vastleggen welke type auto de klant heeft, wat de uitvoering is, wat het kenteken is, wat het bouwjaar is en wat de kleur is. Een klant kan meer dan één auto bezitten.

Maak een database ontwerp (ERD) in [Lucichart](#).

--

# OOP Eindopdracht

In de OOP Challenge moet je een formulier maken en aan de hand van de waarden die je invult moet je dan een berekening maken. Deze berekening laat dan zien welk vervoermiddel sneller is; de fiets of de scooter. Op deze pagina wordt in een stappenplan uitgelegd hoe je dit kunt aanpakken.

Een voorbeeld van hoe de applicatie er zou kunnen uitzien zie je: [hier](#).

[image-1676903058536.png](#)

## Stap 1

Maak in HTML een web form waarin alle invoer velden worden weergegeven. Dit zijn de velden zoals je ze in de voorbeeldapplicatie ziet, dus lengte van de weg, vertraging door file, soort fiets, bandenspanning fiets, soort scooter en bandenspanning scooter.

### Bronnen om te raadplegen

Form (algemeen): [https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)

Radio buttons (type fiets en scooter): [https://www.w3schools.com/tags/att\\_input\\_type\\_radio.asp](https://www.w3schools.com/tags/att_input_type_radio.asp)

In het form geeft je ook aan hoe je het formulier wilt posten, via GET of POST. Weet je niet wat te kiezen, neem dan GET.

## Stap 2

Zorg ervoor dat je het formulier post en dat je op een nieuwe pagina (de pagina waarna je het formulier stuurt), alle variabelen met hun waarden afdruckt.

Je output ziet er dan bijvoorbeeld zo uit:

[image-1676903949706.png](#)

## Stap 3:

Druk nu ook de snelheid van het gekozen voertuig af:

## Stap 4:

Je ziet nu in het bovenstaande voorbeeld dat de scooter 25 km/uur rijdt. Hoe lang doet hij dan over 50 km?

Elk uur rijdt de scooter 25 km, dus hij doet 2 uur over 50 km.

Dit kan je ook in een formule zetten.

```
reistijd = weglengte / standaardSnelheidScooter
```

Als de bandenspanning 50 (%) bedraagt dan doet de scooter er 2x zo lang over, dus in het voorbeeld hierboven doet hij er geen 2 uur maar 4 uur over. We zetten dit ook in de berekening.

```
reistijd = weglengte / standaardSnelheidScooter
```

```
reistijd = reistijd * bandenspannignScooter/100
```

Op regel 2 wordt de (nieuwe) reistijd dus de (oude) reistijd x bandenspanning/100.

Bereken de reistijd voor de scooter én voor de fiets. Gebruik de bovenstaande formules. Druk de waardes af en controleer of alles klopt.

## Stap 5

Je hebt nu de reistijd in uren. Elk uur heeft 60 minuten. Vermenigvuldig de reistijd met 60 zodat je de reistijd in minuten krijgt.

Druk de reistijden van de scooter en fiets nu af in minuten.

Alleen de scooter heeft last van vertraging. Tel het aantal minuten vertraging nu op bij de reistijd van de scooter.

## Stap 6

Je hebt nu twee reistijden. Bepaal welk voertuig het snelste is en geef een reisadvies zoals dat ook in de voorbeeldapplicatie is gedaan.

Je bent nu eigenlijk klaar maar we gaan onze code nu in OOP (objecten) zetten.

## Stap 7, OOP

- Voeg aan het **eind** van je code, code toe die alles in OOP zet.
- Maak de classes en maak (in eerste instantie) public properties in de classes. Elke variabele in je 'oude' code moet ergens in een class als property worden geplaatst.
- Instantieer de objecten en geef alle public properties de juiste waarden. Voor het zetten van de juiste waarden gebruik je de `$_GET['key']`, values, dat zijn de waarden die uit het form komen.
- Voeg een public property *reistijd* toe in de class.
- Zet een method/functie in de class *fiets* en *scooter* waarin je de reistijd **berekent**. Gebruik hiervoor dezelfde formules die je ook in je 'oude' code (uit stap 1 t/m 6) hebt gebruikt.
- Roep de methods/functies aan waarmee de berekening wordt uitgevoerd. De property reistijd van scooter en van fiets worden nu door het gebruik van de method/functie in de class op de juiste waarde gezet.

Druk deze properties af en vergelijk of de waarden kloppen aan de hand van je 'oude' code van stap 1 t/m 6

- Deze reistijden (properties) kun je nu gebruiken om een reisadvies te geven. Werkt alles dan haal je al je code die je in stap 1 tot en met stap 6 hebt gemaakt weg.

## Stap 8 (optioneel)

Je kunt nu de properties van je classes private maken en voor het zetten en uitlezen van de waarden *setters* en *getters* gebruiken.

## Stap 9, vormgeving, testen en commentaar

Maak alles af. Zorg dat alles er netjes uitziet. Let op de **vormgeving** (CSS) en **test** alles goed. Plaats nog wat **commentaar** in je code zodat anderen (en jij later) ook begrijpen wat je doet.

--

[roc.ovh/link/655](https://roc.ovh/link/655)





# Groep 8 - Introductie web site maken

*We gaan een eenvoudige website maken en gebruiken Notepad en de Chrome browser.*

*We gaan kennis maken met de technieken HTML, JavaScript en CSS.*

*We leren wat een variabele in een programmeertaal is.*

## Intro

(presentatie)

Intro en voorbeeld van wat we gaan maken.

Na 10 weken: <http://www.softwaredeveloper.ovh/max/portfolio-samuel/>

Wij gaan een eenvoudige versie hiervan maken:

[image-1676980318643.png](#)

*Aan de slag (eerst individueel en dan je buurman/vrouw helpen).*

## Vorbereiding

(modellen)

Maak nieuw bestand op desktop, **site.txt** en verander naam in **site.html** -> hoe op een Chromebook??

....

## HTML

(modellen)

HTML boilerplate (van internet) en eenvoudige Javascript.

Zoek in Google naar HTML Boilerplate (freecodecamp.org)

1. `<script>....</script>` weghalen.
2. `<link>` regel weghalen.
3. Bedenk een eigen titel (tussen de `<title>` en `</title>`
4. In de body: `<h2>Welkom op mijn site</h2>`

Testen en elkaar helpen.

# Programmeren

(uitleg/presentatie)

(eerst uitleg over variabele -> PowerPoint).

Uitleg variabele -> powerpoint analogie met telefoonnummer en doosje.

(modellen)

## Java Script

```
<script>
  $naam = prompt("Hi, hoe heet jij?");
</script>
```

```
<script>
  naam = prompt("Hi, hoe heet jij?");
  document.write("<h2>Welkom op mijn website "+naam+"</h2>");
</script>
```

(de andere welkom-boodschap mag nu weg)

# Plaatje

(modellen)

Plaatje downloaden (of link kopiëren) en toevoegen (optioneel)

- Zoek in Google (plaatjes) met de term `cartoon character` en Download een plaatje!

In <body>

```

```

of

```

```

In Chrome: **Ctrl-shift-I** en dan **console** voor foutmeldingen.

## Opmaak (CSS)

(modellen)

Zoek in Google naar **css color examples** (w3.org)

```
<style>  
body {  
  background-color:beige;  
}  
h2 {  
  color:crimson;  
}  
</style>
```

## Afsluiten

- Web pagina bestaat uit HTML en CSS. HTML geeft de **basis-structuur** aan en CSS is voor de **vorm/kleuren**.
- **JavaScript** is een programmeertaal die je kan gebruiken voor je website.
- We hebben geleerd wat een **variabele** is.

--

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Mijn Web Site</title>
<script>
  naam = prompt("Hi, hoe heet jij?");
  document.write("<h2>Welkom op mijn website "+naam+"</h2>");
</script>
<style>
  body {
    background-color: beige;
  }
  h2 {
    color: crimson;
  }
</style>
</head>
<body>

</body>
</html>
```

# Yii installeren in live omgeving

Je hebt je site live gezet en je hebt je Yii project (bijvoorbeeld) via FTP overgezet.

Je gaat naar jouwsite/yiiprojectfolder/web en je krijgt een foutmelding?

## Probleem

Bij het overzetten van een Yii project naar *jouwsite* kan je een foutmelding verwachten.

In de foutmelding staat dat Composer een PHP versie > 8.x nodig heeft.

Omdat je waarschijnlijk jouw Yii project hebt geïnstalleerd zonder Github, heb je geen compser nodig om de packages te updaten.

We 'hacken' de foutmelding eruit door de onderstaande file aan te passen.

## Oplossing

Aanpassen

YiiDirectory/vendor/composer/platform\_check.php

[image.1681992118466.png](#)

---

# Stoplicht

## Beste student,

Jullie gaan een intake opdracht maken. Volg de stappen hieronder:

1. Download een editor  
[Windows](#) - [Mac](#)
2. Download [stoplicht.htm](#).

## Opdracht 1

Voor de eerste opdracht open je het bestand dat je bij [2](#) hebt gedownload en ga je dit aanpassen.

Alle stappen staan beschreven in de opdracht. Lees het goed door en als je er niet goed uitkomt, lees het dan nog een keer door. Natuurlijk mag je ook dingen op internet opzoeken.

De instructies voor deze opdracht staan hier: [Stoplicht 01.pdf](#)

## Opdracht 2

De tweede opdracht is wat lastiger. Niet alles staat in de opdracht en sommige zaken zul je moeten proberen uit te vinden met behulp van internet. Lukt niet alles 100%, prima maar probeer zover mogelijk te komen!

De instructies voor deze opdracht staan hier: [Stoplicht 02.pdf](#)

Succes!

--

# 10-tallig stelsel

Wist je dat een computer niet eens 'gewoon' tot 10 kan tellen?

Wij hebben tien tekens om getallen aan te geven 0,1,2,3,4,5,6,7,8,9. De positie van deze cijfers bepaald hun waarde.

Want de 9 in 90 is wat anders dan de 9 in 19. In 90 staat de 9 voor  $10 \times 9$  en in 19 staat de negen voor 9 eenheden.

Ander voorbeeld:

237 is samengesteld uit:

$$2 \times 100 = 2 \times 10^2$$

$$3 \times 10 = 3 \times 10^1$$

$$7 \times 1 = 7 \times 10^0$$

$10^2$  betekent 10 tot de tweede macht, dat is  $10 \times 10$  of als je het makkelijk wilt onthouden een 1 met 2 nullen.

$10^6$  betekent 10 tot de zesde macht, dat is  $10 \times 10 \times 10 \times 10 \times 10 \times 10$  of als je het makkelijk wilt onthouden een 1 met 6 nullen.

De 10 komt telkens terug; we hebben 10 cijfers en we gebruiken machten van 10.

Wij mensen rekenen in het 10-tallig stelsel, weet je waarom? Precies omdat we 10 vingers hebben.

Computers hebben geen 10 vingers. Die kennen alleen de stand stroom of geen stroom, aan of uit. Je zou kunnen zeggen die hebben maar twee vingers.

Computers werken dus in het 2-tallig stelsel.

Het tweetallig stelsel heeft maar twee tekens; 0 en 1.

Wow, hoe dan?

Stel we hebben 101

Dan betekent dit

$$1 \times 4 = 1 \times 2^2$$

$$0 \times 2 = 0 \times 2^1$$

$$1 \times 1 = 1 \times 2^0$$

XXXXX



# Concreet of niet - Versie A

**Opdrachtgever:** InnovatieLab BV

## **Projectomschrijving:**

Het doel van dit project is het creëren van een digitale ruimte die onze innovatieve geest en vooruitstrevende projecten weerspiegelt. We willen een platform dat niet alleen informatief is, maar ook interactief, waardoor bezoekers kunnen deelnemen aan een dialoog over technologische vooruitgang en innovatie.

## **Specifieke Eisen:**

- De website moet 'modern' en 'gebruiksvriendelijk' zijn, met een 'intuïtieve navigatie'.
- Er moet 'iets interactiefs' zijn, zoals een forum of een manier voor gebruikers om feedback te geven.
- We willen 'iets unieks' dat onze website onderscheidt van concurrenten, maar we zijn nog niet zeker wat dat zou moeten zijn.
- De site moet 'goed zichtbaar' zijn op alle apparaten, van desktops tot mobiele telefoons.
- 'Beveiliging' is een prioriteit, maar specifieke details zijn nog niet vastgesteld.

## **Doelstellingen:**

- Het verhogen van de zichtbaarheid en toegankelijkheid van onze projecten en initiatieven.
- Het faciliteren van interactie met ons doelpubliek op een 'innovatieve' manier.
- Het versterken van ons merkimago als een leider in technologische innovatie.

## **Tijdsbestek en Budget:**

- Het project moet 'zo snel mogelijk' worden voltooid, maar er is geen specifieke deadline gegeven.
- Het budget is 'flexibel', afhankelijk van de voorgestelde oplossingen en ideeën.

# Concreet of niet - versie B

**Projectnaam:** Digitale Innovatie Hub

**Opdrachtgever:** InnovatieLab BV

## **Projectachtergrond:**

InnovatieLab BV streeft ernaar om een centrale hub te zijn voor technologische innovatie en samenwerking. Ons doel is om een platform te bieden waarop individuen en organisaties kunnen leren over, bijdragen aan, en samenwerken aan baanbrekende projecten in diverse technologische domeinen.

## **Projectomschrijving:**

Het doel van dit project is de ontwikkeling en lancering van een responsive website die als centrale hub dient voor informatie, samenwerking en community engagement rondom technologische innovatie. De website moet de volgende kernfunctionaliteiten bieden:

1. **Informatieve Pagina's:** Gedetailleerde pagina's over onze projecten, diensten, en innovatiegebieden, inclusief case studies en whitepapers.
2. **Interactieve Community Forum:** Een forum voor gebruikers om ideeën uit te wisselen, vragen te stellen, en samen te werken aan projecten.
3. **Gebruikersregistratie en -beheer:** Mogelijkheid voor bezoekers om accounts aan te maken, waarmee ze kunnen deelnemen aan het forum, zich inschrijven voor nieuwsbrieven, en toegang krijgen tot exclusieve content.
4. **Evenementenkalender:** Een overzicht van aankomende evenementen, workshops, en webinars, met de mogelijkheid voor gebruikers om zich in te schrijven.
5. **Contactpagina:** Een pagina met contactinformatie, een contactformulier, en een optie om direct te chatten met een vertegenwoordiger van InnovatieLab.

## **Technische Specificaties:**

- De website moet worden ontwikkeld met een content management systeem (CMS) zoals WordPress of Drupal voor eenvoudige inhoudsbeheer.
- Het ontwerp moet responsive zijn, geoptimaliseerd voor alle apparaten en schermgroottes.
- De website moet voldoen aan de WCAG 2.1 AA richtlijnen voor toegankelijkheid.

- Gebruik van beveiligingsbest practices zoals HTTPS, data-encryptie, en regelmatige security audits.
- Integratie met Google Analytics voor het monitoren van websiteverkeer en gebruikersgedrag.

**Doelstellingen:**

- Lancering van de website binnen 6 maanden na projectstart.
- Bereiken van een maandelijkse bezoekersaantal van minimaal 10.000 unieke bezoekers binnen het eerste jaar na lancering.
- Registratie van ten minste 1.000 gebruikers op het communityforum binnen de eerste 3 maanden na lancering.

**Budget:**

Het totale budget voor dit project is vastgesteld op €50.000, inclusief alle ontwikkelings-, ontwerp-, en lanceringskosten.

**Tijdsbestek:**

De deadline voor de voltooiing van dit project is 6 maanden vanaf de startdatum. Een gedetailleerde projectplanning zal worden opgesteld in overleg met de geselecteerde leverancier.

# Extra Opgave DB1

## Opgave 1A - Student

Maak in Lucidchart de entiteit student. Een student heeft een studentnummer, voornaam, achternaam, adres, huisnummer, postcode, woonplaats en geboortedatum. Het studentnummer is de *primary key* en de tabel heeft geen *foreign key*.

## Opgave 1B - Leerling

Ontwerp in een diagramtool zoals Lucidchart de entiteit 'leerling'. Deze entiteit bevat de volgende attributen: leerlingnummer, voornaam, achternaam, straatnaam, huisnummer, postcode, stad en geboortedatum. Het leerlingnummer dient als de primaire sleutel. Er zijn geen vreemde sleutels aanwezig in deze tabel.

## Opgave 2A - naschoolse activiteiten

In een database worden leerlingen geregistreerd die op de inschrijflijst staan voor naschoolse activiteiten.

Een inschrijflijst kan dus gegevens van meerdere leerlingen bevatten, maar een leerling kan niet op meerdere inschrijflijsten tegelijk staan.

1. Voeg het attribuut (PK) toe achter het juiste kenmerk in de entiteit 'leerling';
2. Maak de correcte relatie tussen de twee entiteiten. Plaats het 'vorkje' aan de zijde met 'veel' (many) om aan te geven waar meerdere instanties mogelijk zijn.
3. De relatie koppelt de PK aan een FK. Identificeer welke FK noodzakelijk is en plaats (FK) achter de juiste FK in de entiteit die verwijst naar 'leerling'.

## Opgave 2B - medische behandeling

In de database worden patiënten geregistreerd die op de wachtlijst staan voor een specifieke medische behandeling.

Elke wachtlijst bevat gegevens van meerdere patiënten, maar een patiënt kan zich niet voor dezelfde behandeling op meerdere wachtlijsten bevinden.

1. Plaats de aanduiding (PK) achter het juiste kenmerk in de entiteit 'patiënt';
2. Teken de correcte associatie tussen de twee entiteiten. Positioneer het 'vele' einde van de relatie (aangeduid met een 'vorkje') bij de entiteit die meerdere instanties kan bevatten.
3. De relatie linkt de PK aan een FK. Bepaal welke FK vereist is en voeg (FK) achter het juiste kenmerk in de entiteit die verwijst naar 'patiënt'.

## Opgave 3A - auteurs

In een database die de samenwerking tussen auteurs en hun gepubliceerde werken vastlegt, wordt elke publicatie toegeschreven aan één hoofdauteur. Deze situatie gaat over hoe auteurs en hun werken aan elkaar gekoppeld zijn.

We identificeren de volgende entiteiten:

- Een **werk**.
  - Een werk heeft een **titel**, een **genre**, een **publicatiedatum**, en een doelgroep **leeftijdscategorie**.
- Een **auteur**.
  - Een auteur heeft een **voornaam**, **achternaam**, **geboortedatum**, en **woonplaats**.

Elk werk wordt door één auteur geschreven, terwijl de meeste auteurs aan meerdere werken bijdragen. De relatie tussen auteur en werk is dus 1:N (één op veel).

Houd rekening met de regel: **Harkje = Meer = FK (Vreemde Sleutel)**

### Opdracht:

1. Creëer de 2 entiteiten (met sleutels, attributen en datatypen) en
2. teken de relatie tussen de twee entiteiten.
3. Specificeer alle datatypen.
4. Definieer de primaire sleutels en de vreemde sleutels.

Maak het Entiteit-Relatiediagram (ERD) in Lucidchart en lever een schermafbeelding van het ERD in.

# Opgave 3B - Muziekevenementen

In een database die de organisatie van muziekevenementen documenteert, wordt elke evenement geleid door één hoofdorganisator. Deze opzet richt zich op de relatie tussen organisatoren en de evenementen die zij beheren.

We definiëren de volgende entiteiten:

- Een **evenement**.
  - Een evenement heeft een **naam**, een **genre**, een **datum**, en een minimale **leeftijd** voor toegang.
- Een **organisator**.
  - Een organisator heeft een **voornaam**, **achternaam**, **geboortedatum**, en **adres**.

Elk evenement wordt door één organisator georganiseerd, terwijl veel organisatoren verantwoordelijk kunnen zijn voor meerdere evenementen. De relatie tussen organisator en evenement is dus 1:N (één op veel).

Houd in gedachten: **Harkje = Meer = FK (Foreign Key)**

## Opdracht:

1. Stel de 2 entiteiten samen (inclusief sleutels, attributen en datatypes) en
2. Schets de relatie tussen de twee entiteiten.
3. Bepaal alle datatypes.
4. Identificeer de primaire sleutels en de vreemde sleutels.

Ontwerp het Entiteit-Relatiediagram (ERD) in een tool zoals Lucidchart en lever een screenshot van het ERD in.

# Opgave 4A - Tuincentrum

Een tuincentrum wil bijhouden van welke klant een plantenbestelling is. Eén klant kan meerdere bestellingen plaatsen. Het omgekeerde is niet mogelijk; één bestelling kan niet bij meerdere klanten horen.

In het scenario hieronder zijn de entiteiten vetgedrukt en de attributen cursief.

Van de bestelling wil je de volgende attributen vastleggen: *bestelnummer, besteldatum, totaalprijs*.

Van de klant wil je de volgende gegevens registreren: *voornaam, achternaam, telefoonnummer, e-mailadres, woonadres*.

Daarnaast wil het tuincentrum bijhouden welke producten er in elke bestelling zitten. Een bestelling kan meerdere producten bevatten en één product kan in meerdere bestellingen voorkomen.

Van een product wil het tuincentrum de volgende informatie vastleggen: *productnaam, prijs per stuk, voorraad, type plant*.

## Opdracht:

1. Creëer de 3 entiteiten (bestelling, klant en product met alle attributen).
2. Duid aan wat de primaire sleutels zijn.
3. Illustreer de relatie tussen klant en bestelling en duid aan wat de vreemde sleutel is.
4. Teken de relatie tussen bestelling en product en geef aan wat de vreemde sleutel is.

Zorg dat de 'harkjes' van de relaties correct zijn gepositioneerd om de 'veel' zijden aan te geven.

## Opgave 4B - Autodealer

Een autodealer wil bijhouden van wie een bepaald voertuig is. Een klant kan eigenaar zijn van meerdere voertuigen, maar een voertuig kan niet meerdere eigenaren hebben.

In het verhaal hieronder zijn de entiteiten met onderstreping aangeduid en de attributen in cursief.

Van het voertuig wil je de volgende attributen vastleggen: *kenteken, merk, model, kleur*.

Van de klant wil je de volgende gegevens registreren: *klantnummer, voornaam, achternaam, telefoonnummer, adres, stad*.

Verder wil de autodealer bijhouden welke onderhoudsbeurten aan een voertuig zijn uitgevoerd. Elk voertuig kan meerdere onderhoudsbeurten hebben ondergaan, en elke onderhoudsbeurt heeft betrekking op slechts één voertuig.

Van een onderhoudsbeurt wil de autodealer de volgende informatie registreren: *datum van onderhoud, aantal gewerkte uren, kosten, en een indicatie of de rekening is voldaan en of het voertuig al is opgehaald*.

Tip: de indicatie kan worden vastgelegd als een integer (1=ja en 0=nee).

## Opdracht:

1. Maak de 3 entiteiten (*voertuig*, *klant* en *onderhoudsbeurt* met alle attributen).
2. Geef aan wat de primaire sleutels zijn.
3. Teken de relatie tussen *klant* en *voertuig* en geef aan wat de vreemde sleutel is.
4. Teken de relatie tussen *onderhoudsbeurt* en *voertuig* en geef aan wat de vreemde sleutel is.

Zorg dat de 'harkjes' van de relaties aan de juiste zijde staan om de 'veel' zijden aan te duiden.

## Opgave 5a - Boekhandel

Een boekhandel wil bijhouden welke klant welk boek heeft gekocht. Een klant kan meerdere boeken kopen, maar elk boek wordt als een uniek exemplaar beschouwd en kan dus slechts één eigenaar hebben op een bepaald moment.

In het scenario hieronder zijn de entiteiten dikgedrukt en de attributen cursief.

Van het boek wil je de volgende attributen vastleggen: **ISBN**, **titel**, **auteur**, **genre**, **prijs**.

Van de klant wil je de volgende gegevens registreren: **klant-ID**, **voornaam**, **achternaam**, **e-mailadres**, **telefoonnummer**.

Daarnaast wil de boekhandel bijhouden welke bestellingen er zijn geplaatst. Een bestelling kan meerdere boeken bevatten, en elk boek kan in verschillende bestellingen voorkomen (als unieke exemplaren).

Van een bestelling wil de boekhandel de volgende informatie vastleggen: **bestelnummer**, **besteldatum**, **totaalbedrag**.

Ten slotte wil de boekhandel de voorraadstatus van elk boek bijhouden. Voor elk boek in de voorraad wil de boekhandel weten: **voorraad-ID**, **ISBN (van het boek)**, **huidige voorraad**, **minimale voorraad**.

## Opdracht:

1. Creëer de 4 entiteiten (**Boek**, **Klant**, **Bestelling**, en **Voorraad** met alle attributen).
2. Duid aan wat de primaire sleutels zijn.
3. Teken de relatie tussen **Klant** en **Bestelling** en geef aan wat de vreemde sleutel is.



4. Illustreer de relatie tussen **Bestelling** en **Boek** via **Voorraad** en duid aan wat de vreemde sleutels zijn.
5. Teken de relatie tussen **Boek** en **Voorraad** en geef aan wat de vreemde sleutel is.

Zorg dat de 'harkjes' van de relaties correct zijn gepositioneerd om de 'veel' zijden aan te duiden, waarbij rekening wordt gehouden met de unieke en meervoudige relaties tussen de entiteiten.

## Opgave 6B - Sportvereniging

Een sportvereniging wil bijhouden welk lid aan welke sport deelneemt. Een lid kan aan meerdere sporten deelnemen, maar binnen één sportdiscipline wordt elk lid uniek geïdentificeerd.

In het verhaal hieronder zijn de entiteiten vetgedrukt en de attributen cursief.

Van de sport wil je de volgende attributen vastleggen: **sport-ID**, **naam**, **soort**, **locatie**, **trainingsdagen**.

Van het lid wil je de volgende gegevens registreren: **lidnummer**, **voornaam**, **achternaam**, **geboortedatum**, **e-mailadres**, **telefoonnummer**.

Daarnaast wil de sportvereniging bijhouden welke evenementen er georganiseerd worden. Een evenement kan meerdere sporten omvatten, en aan elke sport kunnen meerdere evenementen gekoppeld zijn.

Van een evenement wil de sportvereniging de volgende informatie vastleggen: **evenement-ID**, **evenementnaam**, **datum**, **locatie**, **beschrijving**.

Ten slotte wil de sportvereniging de deelname van leden aan evenementen bijhouden. Voor elke deelname willen ze weten: **deelname-ID**, **lidnummer (van het deelnemende lid)**, **evenement-ID (van het evenement)**, **deelnamerol**, **resultaten**.

### Opdracht:

1. Creëer de 4 entiteiten (**Sport**, **Lid**, **Evenement**, en **Deelname** met alle attributen).
2. Bepaal wat de primaire sleutels zijn.
3. Teken de relatie tussen **Lid** en **Sport** en geef aan wat de vreemde sleutel is.
4. Illustreer de relatie tussen **Sport** en **Evenement** en duid aan wat de vreemde sleutels zijn.
5. Teken de relatie tussen **Lid** en **Deelname** en geef aan wat de vreemde sleutel is, en doe hetzelfde voor de relatie tussen **Deelname** en **Evenement**.

Zorg ervoor dat de 'harkjes' van de relaties correct zijn gepositioneerd om de 'veel' zijden aan te duiden, en houd rekening met zowel de unieke als de meervoudige relaties tussen de entiteiten.

--

# JS Challenge - Stappenplan

Hier wordt stap voor stap beschreven hoe je de JS Challenge "Bestelsysteem kantine" kunt uitvoeren.

Op het eerste gezicht lijkt het een hele college opdracht, maar als we het in kleine stapjes opdelen dan valt het wel mee.

De oplossing zoals hier beschreven is een oplossing, er zijn meerdere oplossingen nodig.

Laten we snel beginnen.

## Stap 0

Lees de opdracht goed door en bekijk het filmpje. Ga pas door als je goed snapt wat er moet worden gebouwd.

## Stap 1, GUI (HTML, CSS)

Maak de GUI zo goed mogelijk na van het voorbeeld (filmpje) gebruik HTML en CSS. Dus nog geen JS.

Maak in beide blokjes gebruik van een `<table>` met drie kolommen.

Maak voor elk artikel uit de menulijst een bijbehorende regel in de bestellingslijst.

Artikel vul je in, aantal en prijs laat je (voorlopig) leeg.

[image 1712162393897.png](#)

## Stap 2, Knop maken

Maak in de menulijst ergens een (tijdelijk) knopje. Roep de JS-functie `updateBestelling()` aan. Deze functie zal straks de Bestelling (rechter blokje gaan updaten).

### ChatGPT prompt

Create a HTML button that calls the JS function `updateBestelling()`.

Breidt de JS-functie uit zodat deze de parameters id, artikel, aantal en prijs meekrijgt.

#### ChatGPT prompt

Change the JS function so that an id, artikel, aantal and price can be passed as a parameter.

Verander de aanroep van de functie vanuit het knopje en naar bijvoorbeeld:

```
updateBestelling('cola', 'cola', '1', '€ 2.25');
```

Verander de functie en benoem de vier variabelen die zijn toegevoegd:

```
function updateBestelling(id, artikel, aantal, prijs) {
```

Druk de variabelen af via console.log() en test of alles werkt door via de knop deze functie aan te roepen.

Ga pas naar de volgende stap als alles is getest en werkt.

## Stap 3, de waarden in de bestelling zetten

Geef de <tr> in het Bestelling-blokje de juiste id's.

Dus de eerste regel krijgt id='cola', de tweede id = 'broodjeKaas', etc. etc.

#### ChatGPT prompt

Use the JS function updateBestelling(id, artikel, aantal, prijs) to change the inner HTML of a tr. The inner html needs to become <td>artikel</td><td>aantal</td><td>prijs</td>  
Change the tr with the id that matches the id passed as a parameter.

Controleer of het werkt en of de juiste regel in het rechter blokje wordt aangepast.

Ga pas naar de volgende stap als alles is getest en werkt.

## Stap 4, JS functie clicked()

Roep op de eerste <td> op elke regel in het linker blok een JS-functie aan die heet *clicked()*. Geef als parameter *regel1*, *regel2*, *regel3*, etc mee. Geef elke <tr>, dus regel het id *regel1*, *regel2*, *regel3*, etc.

Dus elke regel, `<tr>` in het rechter blokje heeft nu een id, `regel1`, `regel2`, `regel3`, etc. En op de eerste kolom, `<td>` wordt de JS functie `clicked('regel1')`, `clicked('regel2')`, `clicked('regel3')` aangeroepen.

### ChatGPT prompt

Create a JS function `clicked(id)`. The function looks up a TR with the `id=id`, and then it extracts the inner html value of the first, second and third td. The variables used to store these values are, `artikel`, `aantal`, `prijs`.

Druk de variabelen `artikel`, `aantal` en `prijs` af via een `console.log` en test de functie.

Ga pas naar de volgende stap als alles is getest en werkt.

## Stap 5, van `clicked()` naar `updateBestelling()`

We kunnen nu vanaf de functie `clicked()` de functie `updateBestelling()` aanroepen.

De waarde van het artikel is hetzelfde als het id van de `<tr>` die we in het rechter blokje moeten updaten dus de aanroep wordt:

```
updateBestelling(artikel, artikel, aantal, prijs);
```

Ga pas naar de volgende stap als alles is getest en werkt.

## Stap 6, formattering bijwerking.

In de `aantal` kolom staat nu een getal; het aantal.

Stel je roept `updateBestelling('cola', 'cola', 3, 2.25)`, dan zie je nu in het rechter blokje cola, 3 en 3.25.

### Kolom aantal

In de kolom 'aantal' moet `3 x (á 2.25)` komen te staan.

### ChatGPT prompt

In JS, I have the variables `aantal` and `prijs`. I want to get the string `aantal x (á prijs)`.

Zet in de kolom prijs een Euro-teken voor de prijs.

Ga pas naar de volgende stap als alles is getest en werkt.

## Stap 7, hide regels met 0

Als het *aantal* in de functie *updateBestelling* 0 is, zet dan de tr met het juiste id op *hidden*.

Dus *updateBestelling* ('cola', 'cola', 0, 2.25), zorgt ervoor dat de tr met het id cola op hidden wordt gezet.

Vergeet niet dat als de waarde niet (meer) 0 is dat dan de tr op visible wordt gezet.

Ga pas naar de volgende stap als alles is getest en werkt.

## Stap 8, knopjes weghalen en updaten on change

Zet op de kolom aantal een *oninput* of een *onchange* en roep daarmee de functie *clicked()* aan.

Zoek zelf even uit, evt met ChatGPT, hoe dit werkt.

Haal alle 'onnodige' code weg, zoals *console.log()* en de *onclick* op de eerste kolom.

Zet commentaar in je JS-code zodat je later nog begrijpt hoe het werkt.

Test alles en als het goed is, is alles klaar!

--

# Authenticator app op een nieuwe telefoon activeren.

- Ga naar nl en log in met je schoolaccount.

image-1724673897492.png

- Naar je account (icoon rechtsboven)
- Naar Beveiligingsgegevens (<https://mysignins.microsoft.com/security-info>)

image-1724673927634.png

- Een methode kiezen
- Vervolg stappen op de Authenticator-app op je telefoon.

# Namen en Alliteraties

## A

1. Avontuurlijke Aisha
2. Aardige Ahmed
3. Actieve Ananya
4. Ambitieuze Amir

## B

1. Beste Bilal
2. Blije Bao
3. Behulpzame Bina
4. Boeiende Bashir

## C

1. Coole Carlos
2. Charmante Chandra
3. Creatieve Chen
4. Curieuze Carmen

## D

1. Dappere Dalia
2. Dynamische Diego
3. Doortastende Deepak
4. Dromerige Disha

## E

1. Energieke Elena



2. Enthousiaste Emre
3. Elegante Ewa
4. Eerlijke Ehsan

## **F**

1. Flinke Fatima
2. Fantastische Farid
3. Fijne Femi
4. Flexibele Farah

## **G**

1. Goede Gabriela
2. Grappige Goran
3. Gelukkige Gita
4. Gulzige Gopal

## **H**

1. Hulpvaardige Hana
2. Humoristische Hassan
3. Hartelijke Hina
4. Heldhaftige Haruto

## **I**

1. Inspirerende Imani
2. Intelligente Ibrahim
3. Interessante Indira
4. Inventieve Iqbal

## **J**

1. Juiste Jamila

2. Jodelende Juan
3. Jonge Jaya
4. Jarige Jafar

## **K**

1. Knappe Kiran
2. Krachtige Karim
3. Koddige Keiko
4. Kalme Kofi

## **L**

1. LuieLeila
2. Leuke Luis
3. Lieve Laila
4. Lachende Lian

## **M**

1. Mooie Maya
2. Moedige Malik
3. Mystieke Mei
4. Muzikale Mustafa

## **N**

1. Nette Nadia
2. Nieuwsgierige Nabil
3. Nuchtere Nia
4. Nette Nikhil

## **O**

1. Oprechte Omar

2. Ontspannen Olga
3. Optimistische Oksana
4. Overtuigende Oriana

## **P**

1. Prettige Priya
2. Pientere Pablo
3. Positieve Parvati
4. Prachtige Pasha

## **Q**

1. Quirky Qasim
2. Quasi-quiet Qin
3. Quick-witted Quyen
4. Quaint Qamar

## **R**

1. Rustige Rashida
2. Romantische Rami
3. Rijke Rina
4. Rappe Raj

## **S**

1. Sterke Sara
2. Stralende Sadiq
3. Slimme Sita
4. Sportieve Samir

## **T**

1. Trouwe Tariq

2. Trotse Talia
3. Talentvolle Tarek
4. Tedere Tenzin

## **U**

1. Unieke Usha
2. Uitbundige Usman
3. Uiterst Uyen
4. Uitmuntende Ugo

## **V**

1. Vrolijke Vanya
2. Vindingrijke Vikram
3. Verstandige Violeta
4. Verzorgde Vinh

## **W**

1. Wijsneuzige Wafa
2. Wonderlijke Wei
3. Waardige Walid
4. Warmhartige Wulan

## **X**

1. Extravagante Xander
2. X-factor Ximena
3. Xylofoon-spelende Xia
4. X-traordinary Xiang

## **Y**

1. Yogaliefhebbende Yvonne

2. Jazzy Yuki
3. Jankende Yasmin
4. Yuppie-achtige Yelena

## **Z**

1. Zorgzame Zara
2. Zonnige Zain
3. Zachte Zoya
4. Zeldzame Zubair

--

# Kennismakingsspellen

## Over-de-streep stellingen

1. Als je niet wist wat je vandaag aan moest trekken.
2. Als je gisteren meer dan drie uur hebt gegamed
3. Als je houdt van koken
4. Als je houdt van sporten
5. Als je muziek maakt
6. Als je per week meer dan twee dagen werkt
7. Als je elke dag programmeert
8. Als je wel eens verliefd (bent) geweest
9. Als je een relatie hebt
10. Als je heel erg gemotiveerd bent voor deze opleiding
11. Als je iets ontzettend grappigs wilt delen wat je vandaag hebt beleefd.
12. Als je afgelopen week iets hebt gedaan waar je trots op bent
13. Als je het met iemand deelt als je je happy voelt.
14. Als je wel eens enorm om jezelf hebt gelachen.
15. Als je je wel eens inzet voor een goed doel.
16. Als je wel eens hebt gevochten.
17. Als je heel trots bent op iets van jezelf.
18. Als je je wel eens anders hebt voorgedaan dan wie je werkelijk bent.
19. Als je vindt dat het belangrijk is dat mensen zich vrijwillig voor anderen inzetten.
20. Als je iemand kent die jou inspireerde.
21. Als je wel eens twijfelt aan jezelf.
22. Als je gevoelig bent voor de mening van anderen.
23. Als je het moeilijk vindt om contact te maken met anderen.

24. Als je iemand kent die door problemen alcohol of drugs is gaan gebruiken.
25. Als je iemand in je leven hebt met wie je iets persoonlijks kan delen.
26. Als je iets goeds hebt gedaan voor iemand die je eigenlijk niet mocht.
27. Als je iets met iemand gedeeld hebt dat je eigenlijk voor jezelf had willen houden.
28. Als je eens iets gekregen hebt waar je niet blij mee was.
29. Als je je wel eens buitengesloten hebt gevoeld.
30. Als je met vrienden zou delen als je het niet meer ziet zitten op school.
31. Als je vindt dat delen mooier is dan weggeven.
32. Als je wel eens angstig wordt als je ziet wat er in de wereld gebeurt.

## Stellingen

1. Ik kom altijd gemotiveerd naar school.
2. Mobieltjes in de klas moet verboden worden.
3. Docenten moeten huiswerk controleren anders kom ik in de verleiding om het niet te maken.
4. Whatsappen met docenten moet verboden worden.
5. Mbo-opleidingen moeten zich geheel richten op het opleiden voor het vak en bredere vorming achterwege laten (dus géén Nederlands/Rekenen/Engels/Sport/Burgerschap/LOB).
6. Studenten mogen zelf beslissen welke feestdagen zij willen vieren.
7. Wiskunde moet een verplicht vak worden op het MBO.
8. Jongeren moeten ook kunnen stemmen, daarom moet de kiesgerechtigde leeftijd worden verlaagd van 18 naar 16 jaar.
9. De overheid moet de prijs van fastfood verhogen.
10. Energiedrankjes moeten verboden worden.
11. Supermarkten mogen alleen gezond eten in de aanbieding doen.

## Handstelling

1. **duim:** waar ben je goed in?

2. **wijsvinger:** waar wil je naar toe? Wat zijn je ambities?
3. **middelvinger:** waar heb je een hekel aan?
4. **ringvinger:** waar ben je trouw aan?
5. **pink:** waarin moet je nog groeien, waar ben je nog niet zo goed in?



# (Cyber) Security Overview

## Incident Response Lifecycle (NIST Framework)

The **NIST (National Institute of Standards and Technology)** defines a **6-phase** approach:

### 1. Preparation

- Develop an incident response plan (IRP).
- Train teams, conduct security awareness programs.
- Implement security controls like firewalls, IDS/IPS.

### 2. Identification

- Detect anomalies through logs, SIEM tools, and alerts.
- Confirm if an incident has occurred (malware, data breach, DDoS).
- Classify the severity and type of attack.

### 3. Containment

- Stop the attack from spreading (isolate infected systems).
- Implement temporary patches, disable compromised accounts.
- Ensure forensic evidence is preserved.

### 4. Eradication

- Remove the root cause of the incident (malware, vulnerabilities).
- Apply permanent fixes, patch vulnerabilities.
- Strengthen security measures.

### 5. Recovery

- Restore affected systems from clean backups.
- Monitor for lingering threats, ensure full functionality.
- Reintegrate systems into production.

### 6. Lessons Learned

- Conduct a post-mortem analysis of the incident.
- Document findings, update response plans.
- Improve security policies and employee training.

# Threat Intelligence

(from incident to problem)

- reporting
- monitor new trends and techniques
- operational response: up-to-date (virus) protection
- research into new techniques.

## Example Use Case

- A **financial institution** uses threat intelligence to monitor **new phishing campaigns** targeting its customers.
- The security team blocks malicious domains and updates email filtering rules.
- They prevent thousands of phishing attempts **before customers are impacted**.

# Cryptography

**Cryptography** is the practice of securing information by **converting it into an unreadable format** to prevent unauthorized access. It ensures **confidentiality, integrity, authenticity, and non-repudiation** of data.

## Key Concepts:

- **Encryption** – Converting plaintext into ciphertext (e.g., AES, RSA).
- **Decryption** – Reversing encryption to retrieve the original data.
- **Hashing** – Generating a fixed-length fingerprint of data (e.g., SHA-256).
- **Digital Signatures** – Verifying authenticity using cryptographic keys.
- **Key Exchange** – Securely sharing encryption keys (e.g., Diffie-Hellman).

## Types of Cryptography:

1. **Symmetric Encryption** – Uses one key for both encryption and decryption (e.g., AES).
2. **Asymmetric Encryption** – Uses a public and a private key (e.g., RSA, ECC).

## Uses in Cybersecurity:

- ☑ **Protects data in transit and at rest** (e.g., SSL/TLS for web security).
- ☑ **Secures passwords** (e.g., storing hashes instead of plaintext).
- ☑ **Enables secure authentication** (e.g., digital signatures).
- ☑ **Prevents data tampering** (e.g., message integrity checks).

**Example:** HTTPS encrypts web traffic using SSL/TLS, keeping user data safe from eavesdroppers.

Cryptography is essential for **privacy, secure communication, and data protection** in modern cybersecurity.

# Network

## OSI Model

The **OSI (Open Systems Interconnection) model** is a **7-layer framework** that standardizes communication in a network:

1. **Physical Layer** – Hardware, cables, and wireless signals.
2. **Data Link Layer** – MAC addresses, switches, and error detection.
3. **Network Layer** – IP addressing and routing.
4. **Transport Layer** – Reliable data transmission (TCP, UDP).
5. **Session Layer** – Managing communication sessions.
6. **Presentation Layer** – Encryption, compression, data formats.
7. **Application Layer** – User interactions (HTTP, FTP, DNS).

## Relation to Cybersecurity:

- **Physical Layer** – Preventing physical tampering with devices.
- **Data Link Layer** – Protecting against MAC spoofing and ARP poisoning.
- **Network Layer** – Firewalls, IP filtering, and VPNs.
- **Transport Layer** – Preventing DDoS attacks, ensuring encryption (TLS).
- **Session Layer** – Secure authentication and session hijacking protection.

- **Presentation Layer** - Implementing SSL/TLS encryption.
- **Application Layer** - Web security (XSS, SQL injection, authentication).

# AI en Onderwijs

Samenvatting van debat met [Michiel Bakker](#) (MIT), [Wietse Hage](#) en [Alexander Klöpping](#)

---

## Inleiding

AI transformeert de Arbeidsmarkt en stelt het Onderwijs voor Fundamentele Vragen

De bronnen benadrukken dat AI, met name de ontwikkeling van **Artificial General Intelligence (AGI)**, de arbeidsmarkt radicaal zal veranderen. AI-systemen worden steeds beter in het uitvoeren van cognitieve taken, wat traditionele kennisbanen kan overbodig maken, vergelijkbaar met hoe een gepensioneerde programmeur zich nu al voelt door de snelheid waarmee AI code kan genereren. De spreker, Michiel Bakker, gelooft dat **AGI** binnen 3 tot 4 jaar substantieel aanwezig kan zijn en alle cognitieve taken die mensen kunnen, beter kan uitvoeren.

Het traditionele onderwijsmodel, gebaseerd op het delen van kennis door een docent en toetsen om retentie te meten, is gebouwd op schaarste van kennis. AI maakt kennis echter altijd en overal beschikbaar, en kan zich perfect aanpassen aan individuele leerstijlen. Dit dwingt ons na te denken over wat onderwijs werkelijk is en welke vaardigheden in de toekomst waardevol blijven. De discussie over de impact van AI is fundamenteel anders en sneller dan eerdere technologische revoluties, zoals de komst van de rekenmachine.

## Het Debat: Thick Client vs. Thin Client

Een centrale vraag is hoe "dik" of "dun" mensen (en dus studenten) moeten zijn in termen van kennis en vaardigheden.

- De "thick client" metafoor staat voor een mens met een breed scala aan diepgaande kennis en vaardigheden, die zelfstandig kan functioneren (vergelijkbaar met een piloot die zonder autopilot kan vliegen).
- De "thin client" metafoor staat voor een mens die voornamelijk een interface is, sterk afhankelijk van externe intelligentie (AI/datacenters) voor de benodigde kennis of rekenkracht. Organisaties zouden de voorkeur kunnen geven aan thin clients, omdat personeel dan minder onvervangbaar is en kennis/processen in systemen geborgd zijn. Dit is efficiënter vanuit bedrijfs perspectief.

Als de arbeidsmarkt evolueert naar een model waarin mensen voornamelijk "operators" of "managers van agents"

zijn – systemen die het eigenlijke werk doen – dan zou een "thin client" benadering in het onderwijs functioneel kunnen zijn. Echter, de vraag is of een "dunne" mens wel goed toezicht kan houden op

of de juiste vragen kan stellen aan een veel "slimmer" AI-systeem. Het trainen van mensen voor beroepen die aantoonbaar zullen verdwijnen door automatisering (zoals wellicht bepaalde vormen van tekstvertaling of traditioneel boekhouden) wordt als onverantwoord gezien.

## **Wat moet er veranderen?**

### **Curriculum en Focus:**

- Er moet een verschuiving komen in beroepsonderwijs: minder focus op specifieke, potentieel te automatiseren taken, en meer op vaardigheden om AI-systemen aan te sturen en te managen. Dit 'operator' zijn zou een basale, breed toepasbare vaardigheid kunnen worden.
- Naast technische aansturing van AI, moet het onderwijs zich richten op vaardigheden die AI minder goed kan, zoals creativiteit, empathie, menselijk inzicht, en sociale interactie.
- Kritisch denken en het leren herkennen en evalueren van argumenten blijven cruciaal, mogelijk aangeleerd via AI-systemen die leerlingen begeleiden in plaats van antwoorden te geven.
- Het onderwijs moet mogelijk accepteren dat niet iedereen fulltime aan het werk zal zijn en zich meer richten op menselijke bloei, zelfontwikkeling en maatschappelijk engagement buiten traditionele beroepen.

### **Methoden en Technologiegebruik**

- AI moet breed worden ingezet als hulpmiddel in het leerproces, bijvoorbeeld als gepersonaliseerde tutor. Studenten moeten leren AI-tools effectief te gebruiken, zelfs om hun eigen werk te automatiseren.
- De discussie over AI in het onderwijs moet erkennen dat AI-systemen (zoals taalmodellen) maakbaar en configureerbaar zijn, en niet statisch. AI kan specifiek worden ontworpen voor educatieve doeleinden, bijvoorbeeld door geen directe antwoorden te geven maar de student te begeleiden (het "Socrates-model").
- Het onderwijs moet een balans vinden tussen individueel leren met AI en groepsactiviteiten. Fysieke scholen zouden meer centra kunnen worden voor sociale interactie en groepsleren, terwijl individueel leren deels thuis met AI kan plaatsvinden.
- De rol van de docent verandert: minder pure kennisoverdracht (een routineuze taak die AI kan overnemen) en meer faciliteren, begeleiden, en focussen op het welzijn en de sociaal-emotionele ontwikkeling van studenten.

### **Systeemniveau en Urgentie**

- Er is een dringende noodzaak om het onderwijssysteem aan te passen; er is geen 10 jaar de tijd zoals bij de rekenmachine.
- De discussie over aanpassing verloopt te langzaam en is niet productief genoeg, deels doordat de dynamische aard van AI wordt genegeerd.
- Met name beroepsopleidingen moeten jaarlijks kritisch evalueren of de aangeboden opleidingen nog aansluiten bij de veranderende arbeidsmarkt.
- Oude systemen van categorisatie en toelating op basis van specifieke vakken (zoals een 9 voor wiskunde) kunnen worden vervangen door fijnmazigere assessments mogelijk gemaakt door AI.
- Er is frustratie dat onderwijsinstellingen en bedrijven (met name uitgevers) de kansen om het onderwijs fundamenteel te verbeteren met AI nog onvoldoende benutten. De kosten voor het ontwikkelen van educatieve software zijn lager geworden, wat de barrière voor innovatie zou moeten verlagen.

## Samenvattend

Het onderwijssysteem moet niet alleen de inhoud aanpassen door te focussen op vaardigheden die complementair zijn aan AI (agent management, kritisch denken, sociale/emotionele intelligentie), maar ook de methode door AI te integreren als een flexibel en configureerbaar hulpmiddel voor gepersonaliseerd leren.

De rol van de docent verschuift, en de school als fysieke plek kan meer gericht worden op sociale en groepsleren. Dit alles vereist snelle en fundamentele veranderingen, weg van het verouderde kennis-schaarste model en de artificiële structuren die daaruit voortkwamen.

--

# PDO

## PDO.WIJS.OVH

image.png and or type unknown

### database.php

```
$dsn = 'mysql:host=localhost;dbname=mijn_database;charset=utf8mb4';  
$gebruiker = 'root';  
$wachtwoord = '';
```

### Voer query uit

```
<?php  
// □ Laad de database-gegevens in uit een apart bestand  
include 'database.php';  
  
// □ Maak verbinding met de database met behulp van PDO  
$conn = new PDO($dsn, $gebruiker, $wachtwoord);  
// □ Schrijf de SQL-query met een parameter voor veiligheid  
$sql = "SELECT * FROM studenten WHERE klas = :klas";  
  
// □ Bereid de query voor (voorkomt SQL-injectie)  
$stmt = $conn->prepare($sql);  
  
// □ Koppel de waarde 'ICT1A' aan de :klas-parameter in de query  
$stmt->bindParam(':klas', $klas);  
$klas = 'ICT1A';  
  
// ► Voer de query uit  
$stmt->execute();  
  
// □ Haal alle resultaten op als associatieve array  
$resultaten = $stmt->fetchAll(PDO::FETCH_ASSOC);  
  
// □ Loop door de resultaten en toon de naam van elke student
```



```
foreach ($resultaten as $rij) {  
    echo $rij['naam'] . "<br>";  
}
```

## Insert

```
<?php  
// □ Laad de database-gegevens in uit een apart bestand  
require_once 'database.php';  
  
// □ Maak verbinding met de database met behulp van PDO  
$conn = new PDO($dsn, $gebruiker, $wachtwoord);  
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
  
// □ Schrijf de SQL-query voor het invoegen van een student  
$sql = "INSERT INTO studenten (naam, klas) VALUES (:naam, :klas)";  
  
// □ Bereid de query voor  
$stmt = $conn->prepare($sql);  
  
// □ Geef de waardes op voor de parameters  
$naam = 'Ali';  
$klas = 'ICT1A';  
$stmt->bindParam(':naam', $naam);  
$stmt->bindParam(':klas', $klas);  
  
// ► Voer de query uit  
$stmt->execute();  
  
echo "Student toegevoegd!";  
?>
```