

USABILITY



ONTWIKKELGROEP:

KES GREUTER (UNIVERSITEIT TWENTE)

DR. INGRID BREYMANN (UNIVERSITEIT TWENTE)

DR. IR. JAN-WILLEM VAN 'T KLOOSTER (UNIVERSITEIT TWENTE)

IR. GERRIT JAN DE WILDE MBA (BATAAFS LYCEAM HENGEL)

BEWERKT DOOR HAROLD PRINS (GREIJLDANUS ZWOLLE)

UNIVERSITY
OF TWENTE.

JOUW NAAM: _____

INHOUDSOPGAVE

Inhoudsopgave	1
Inleiding.....	2
Leerdoelen.....	3
Vooraf.....	4
Les 1 Introductie Usability.....	7
 Huiswerk leesopdracht richtlijnen	13
Les 2 Heuristieken	20
Les 3 Introductie HTML	26
Les 4 Introductie CSS	37
Les 5 Flexbox	48
Les 6 Responsive design	51
Les 7 Ontwerp cyclus.....	54
Les 8 Functioneel ontwerp	59
Les 9 Prototype maken.....	71
Les 10 Evaluatie methodes.....	73
Les 10a Subjectieve evaluatie methoden.....	75
Les 10b Objectieve evaluatie methoden.....	85
Onderzoek uitvoeren.....	101
Index	102
Extra html/ css - Eenheden.....	107
Logboek	109

INLEIDING

In deze periode ga je leren wat usability (US) is, en hoe je voor een goede usability kan zorgen. We gaan dat direct toepassen in een digitaal product. Namelijk een website. Daarom staan er in deze module ook html en css lessen. We ronden deze module af met een praktische opdracht die je in tweetallen gaat doen. Daarin ga je een website maken en daarop de testmethodes toepassen.

Het examenprogramma voor het domein 'Usability' bestaat uit subdomeinen 54 t/m 56.

54 Gebruikersinterfaces

De kandidaat kan de werking van gebruikersinterfaces beschrijven en verklaren aan de hand van cognitieve en biologische modellen.

55 Gebruikersonderzoek

De kandidaat kan gebruikersinterfaces van digitale artefacten evalueren via gebruikersonderzoek.

56 Ontwerp

De kandidaat kan elementen van een gebruikersinterface ontwerpen.

Uitleg over dit boek

OPDRACHT # OEFENINGEN

Dit zijn de opdrachten voor in de les of als je dit niet klaar hebt kan je het thuis afmaken. Ze worden de hele module door doorgenummerd.

OPDRACHT # PRAKTISCH ONDERZOEK

Dit is het onderzoeksdeel van de praktische opdrachten. Deze hebben een eigen nummering en worden beantwoord op maken.mijn-in.nl. We maken de praktische opdrachten in tweetallen. Deze delen we de eerste les in. De laatste 2 opdrachten (49,50) lever je in SomToDay in.

OPDRACHT # FUNCTIONEEL ONTWERP

Dit is het functioneel ontwerp, ook deze maken we in tweetallen. Deze opdrachten hebben ook een eigen nummering en worden opgenomen in een apart Google document dat je met mij deelt en inlevert op SomToDay als PDF.

OPDRACHT # HTML EN CSS

En dan hebben we nog de html en css opdrachten. Je zorgt dat je de antwoorden van deze opdrachten goed bewaard. Je werkt ze uit op maken.mijn-in.nl

LEERDOELEN

De leerdoelen voor deze module zijn:

12. Ik begrijp wat usability en user experience zijn, ken de bijbehorende aspecten en kan het verschil ertussen uitleggen.
13. Ik kan de heuristieken van Nielsen en de designprincipes van Dieter Rams benoemen, uitleggen en herkennen in bestaande interfaces.
14. Ik begrijp hoe gebruikersonderzoek werkt, inclusief begrippen als testobject, testbasis, testmethode en iteratief testen.
15. Ik kan verschillende evaluatiemethoden uitleggen en toepassen, zoals Think Aloud, System Usability Scale (SUS), focusgroepen en A/B-testing.
16. Ik kan onderscheid maken tussen subjectieve en objectieve evaluaties en deze op de juiste manier voorbereiden, uitvoeren en resultaten verwerken.
17. Ik kan evaluatiemethodes afwegen en kritisch beoordelen op bruikbaarheid en betrouwbaarheid voor een specifiek onderzoek.
18. Ik kan de heuristieken van Nielsen toepassen op bestaande digitale producten.
19. Ik kan op basis van onderzoek zelf een gebruikersinterface ontwerpen of verbeteren.
20. Ik kan onderzoeksresultaten van mezelf en anderen presenteren en beoordelen.
21. Ik kan een prototype van een website maken met HTML en CSS.

VOORAF

Beste leerling,

Voor je ligt de module Usability. We gaan in het eerste half jaar van klas 4 met deze module aan de slag. Het belangrijkste is dat we een bruikbare website gaan maken. Een website dus waarvan de usability goed is. Daarvoor gaan we vier dingen doen.

1. We gaan ontdekken wat Usability is (en wat niet). Om dat te verwerken gaan we de blauwe opdrachten maken. Deze theorie is de basis voor enkele vragen op de toets in periode 2.

Daarnaast gaan we de vier fases van de ontwerpcyclus doorlopen. Sommigen wat uitgebreider dan anderen. Zie voor meer informatie les 7. Daarvoor beginnen we in de analyse fase.

2. In de analyse fase gaan we een functioneel ontwerp maken voor een website (zie les 8). Dit levert het eerste document voor de Praktische Opdracht op. Dit zijn de groene opdrachten.
3. In de prototype fase gaan we de website daadwerkelijk maken. Daarvoor gaan we eerst de inleidende lessen html en css volgen. In les 10 volgt dan de daadwerkelijke opdracht voor het maken van het prototype. De basis daarvoor is uiteraard het functioneel ontwerp. Dit zijn de gele opdrachten. Dit levert het tweede product op van de PO, namelijk de website.
4. Als laatste gaan we onderzoek doen naar usability. We doen dat deels op een website die je zelf kiest en deels op de website die je maakt in de prototype fase. Dit zijn de oranje opdrachten. Dit levert het 3^e product op namelijk de antwoorden op alle onderzoekopdrachten. Dit laatste document is meteen ook (net als de blauwe opdrachten) een voorbereiding op de eindtoets.

Bij alle opdrachten werken we in tweetallen. Dus de website, het functioneel ontwerp en het onderzoek doe je met z'n tweeën. Je houdt voor jezelf een logboek bij zoals bijgevoegd in de bijlage. Deze lever je met de andere producten in. Daarnaast zorgen jullie ervoor dat je beide het gemaakte werk kan toelichten.

In de beoordeling van de PO zal het functioneel ontwerp en het prototype (de ontwikkelde website) het zwaarst wegen. Het beoordelingsformulier ziet er als volgt uit:

Onderwerp	Punten	
Functioneel ontwerp (35) → les 8		
Noteer in hoofdstuk 1 de doelen van het product en beschrijf de doelgroep.	5	
Geef in hoofdstuk 2 een lijst van eisen en een lijst van wensen.	5	
Hoofdstuk 3 userstories	5	
Hoofdstuk 4 structuur	5	

Schrijf nu Hoofdstuk 5: basiselementen. In dat hoofdstuk zet je neer welke hoofdafbeelding, kleurenstaal en logo je gaat gebruiken. Motiveer je keuzes kort.	5	
Verwerk je schermontwerpen in hoofdstuk 6: schermontwerp. Noteer daarbij voor elk scherm welke ontwerp erop van toepassing is.	5	
o Een goede voorkant o Een inhoudsopgave	5	
Prototype/ Website (45) → les 9		
Voldoet aan functioneel ontwerp	10	
Maakt gebruik van blokken/ grid Juist gebruik van html	5	
Doctype/ head / body	5	
Netjes gecodeerd (leesbaar en juist gebruik inspringing)	10	
Afbeeldingen juist ingevoegd (relative urls)	5	
Responsive	5	
Extra functionaliteit (slider)	5	
Onderzoek Usability (20)		
Usability opdrachten 3, 5 ,10 , 26 op maken.mijn-in.nl	10	
Twee uitgewerkte onderzoeksmethodes, bij voorkeur objectief en subjectief. (één op de gekozen rekensite en één op je eigen ontwikkelde prototype) – opdrachten 49, 50 op SomToDay	10	
GEMIST EERSTE DEADLINE	-20pnt	
SCORE		
CIJFER		

Eerste deadline: _____

Tweede deadline (-20pnt): _____



OPDRACHT 1.

KENNISMAKINGSOPDRACHT

Ter kennismaking vragen we je om in tweetallen kritisch te kijken naar sites, programma's of apps.

Denk na over sites, programma's of apps waar je iets opvalt. Iets dat je bijvoorbeeld heel bijzonder vindt, heel erg leuk of juist erg vervelend of verassend. Kies er drie uit. Maak voor elke keuze een lijst met redenen waarom je deze goed, of juist slecht vindt.

Kies er uit de volgende minimaal 1:

- <https://www.art.yale.edu/>
- <https://www.pennyjuice.com/>
- <https://www.pnwx.com/>
- <https://www.lingscars.com/> of de vorige versie <https://museum.lingscars.com/>

Site: _____

Beoordeling: _____

Redenen:

1. _____

2. _____

3. _____

Site: _____

Beoordeling: _____

Redenen:

1. _____

2. _____

3. _____

Site: _____

Beoordeling: _____

Redenen:

1. _____

2. _____

3. _____

LES 1 INTRODUCTIE USABILITY

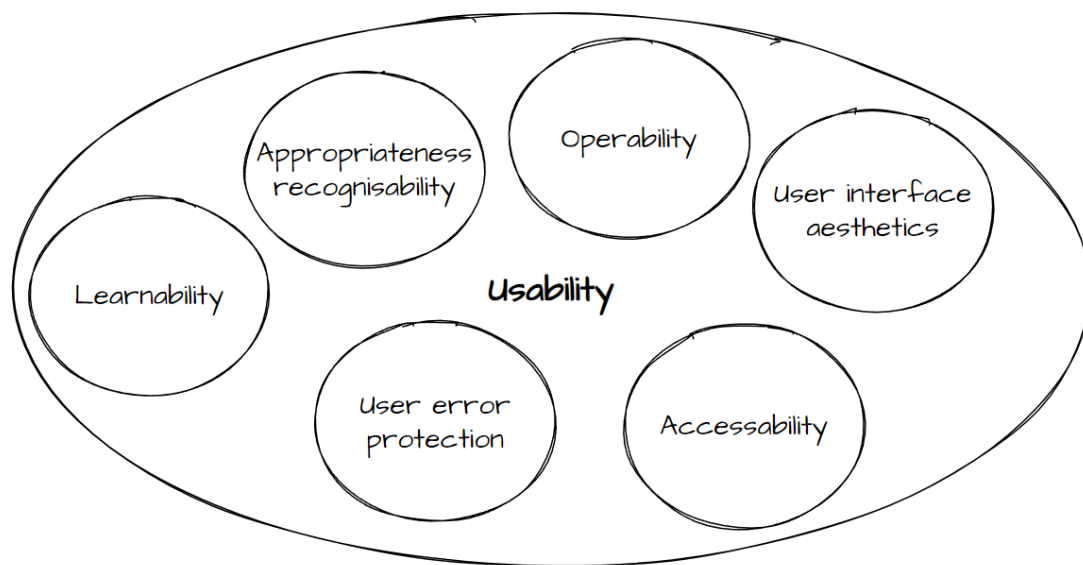
Usability betekent gebruiksvriendelijkheid of bruikbaarheid. Het is een vakgebied dat zich bezig houdt met de gebruiksvriendelijkheid van gemaakte producten. Tijdens de ontwikkeling van producten, machines, gereedschap, processen, websites etc. is het belangrijk om voldoende rekening te houden met de gebruiker van het eindproduct. Hoe toegankelijker en gebruiksvriendelijker een product is, hoe meer mensen het product op een makkelijke manier kunnen gebruiken.

Er zijn veel verschillende definities van usability. Niet alle wetenschappers en onderzoekers zijn het met elkaar eens. Dit is omdat het niet makkelijk is om een definitie te geven voor usability die alle mogelijke aspecten dekt. Voor deze lesmodule is gekozen voor de definitie van de Internationale Organisatie voor Standaardisatie (ISO), dit is namelijk bekende en vaakgebruikte literatuur.

Deze definitie is: *'Usability is de mate waarin een product of systeem gebruikt kan worden door gespecificeerde gebruikers om effectief, efficiënt en naar tevredenheid gespecificeerde doelen te bereiken in een gespecificeerde gebruikscontext.'*

ISO 25010 omschrijft usability preciezer vanuit de volgende aspecten:

1. **Appropriateness of recognisability** (Herkenbaarheid van geschiktheid)
De mate waarin gebruikers kunnen herkennen of een product of systeem geschikt is voor hun behoeften.
2. **Learnability** (Leerbaarheid)
De mate waarin een product of systeem door een bepaalde gebruiker gemakkelijk gebruikt kan worden om een bepaald doel te bereiken zodat het product of systeem efficiënt, risico-vrij en bevredigend gebruikt kan worden in een bepaalde context. Learnability zegt iets over hoe makkelijk het is om een taak voor het eerst aan te leren is en over hoe gemakkelijk het is om deze taak te hierna blijven kunnen doen.
3. **Operability** (Bedienbaarheid)
De mate waarin een product of systeem attributen heeft die het makkelijk maken om het te bedienen.
4. **User error protection** (Voorkomen gebruikersfouten)
De mate waarin het systeem gebruikers beschermt tegen het maken van fouten.
5. **User interface aesthetics** (Volmaaktheid gebruikersinteractie)
De mate waarin een gebruikers interface het de gebruiker mogelijk maakt om een plezierige en bevredigende interactie met het systeem te hebben. De focus ligt hier dus op hoe mooi en plezierig een product is.
6. **Accessibility** (Toegankelijkheid)
De mate waarin een product of systeem gebruikt kan worden door mensen met de meest uiteenlopende kenmerken en bekwaamheden om een gespecificeerd doel te bereiken in een gespecificeerde gebruikscontext.



Figuur 1: De aspecten van usability

OPDRACHT 2. YAHOO!

Beoordeel de voorpagina van Yahoo! (www.yahoo.com) op usability. Kies vier van de zes usability aspecten en beschrijf hoe Yahoo deze bereikt. Gebruik ongeveer 50 woorden per aspect.

Aspect: _____

Reden: _____

Aspect: _____

Reden: _____

Aspect:	_____
Reden:	_____

Aspect:	_____
Reden:	_____

OPDRACHT 3. START PRAKTISCHE OPDRACHT ONDERZOEK

Gelijk met de lessen gaan we aan de slag met de praktische opdracht waarin je onderzoek doet naar de usability van een website. We maken daarbij gebruik van twee websites. De eerste is de website die je zelf ontwikkeld. Zie de groene opdrachten voor het functioneel ontwerp en de gele opdrachten voor het prototype van de website. Daarnaast maken we gebruik van een bestaande website. Beide website houden zich bezig met kinderen op de basisschool.

Je mag kiezen uit de volgende websites die allemaal over rekenen gaan:

- <https://www.fi.uu.nl/rekenweb/>
- <https://www.tafeldiploma.nl/>
- <https://www.beterrekenen.nl/>
- <https://www.junioreinstein.nl/> (deze kun je als spiegel gebruiken)

Wat moet je doen:

- Maak deze vraag op maken.mijn-in.nl.
- Hierin beschrijven jullie één van de gegeven sites (keuze staat hierboven) die je gaat evalueren. Deze beschrijving moet de functie van het product en een link naar het product bevatten. Dit is minimaal 200 woorden.
- Benoem ook 3 aspecten die je aanspreekt en geef er een reden bij (50 woorden per reden).

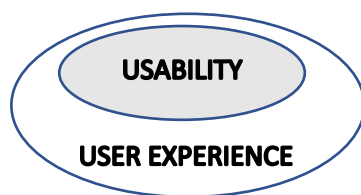
Leesopdracht US vs UX

Deze leesopdracht gaat over het verschil en de overeenkomsten tussen usability en user experience. Lees onderstaande tekst en maak voor jezelf een samenvatting. Deze informatie zul je in les 2 nodig hebben.

Je weet nu dat usability gaat om het creëren van *gebruiksvriendelijkheid* en bruikbaarheid van een product of systeem. Usability is één aspect van user experience (=gebruikerservaring) van een product of systeem. User experience is dus een veel breder begrip. User experience draait om het creëren van een zo goed mogelijke *gebruikerservaring* bij een gebruiker tijdens het gebruik van een app, website, product of dienst.

Volgens ISO 9241 omvat de *gebruikerservaring* alle emoties, overtuigingen, voorkeuren, waarnemingen, fysieke en psychologische reacties, gedrag en prestaties die voor, tijdens en na het gebruik van een product of dienst optreden.

Een goede user experience ontstaat door verschillende factoren, waaronder usability. Het is belangrijk om het verschil tussen user experience en usability goed te weten. Usability focust zich alleen op het behalen van het praktische doel van het gebruik van het product, dus de functionele eisen en de benodigde aesthetics. Goede usability maakt een taak gemakkelijk en intuïtief. Het is daarbij belangrijk om stappen die een gebruiker moet voltooien te minimaliseren en hindernissen te verwijderen. Dat doe je door je af te vragen wat gebruikers doen, en hoe ze het doen. Het waardevol maken van een taak, het scheppen van een emotionele band en de vraag wat gebruikers voelen vallen niet onder usability, maar wel onder user experience.



Figuur 2: Usability is onderdeel van user experience



Figuur 3: De focus van usability en de focus van user experience.

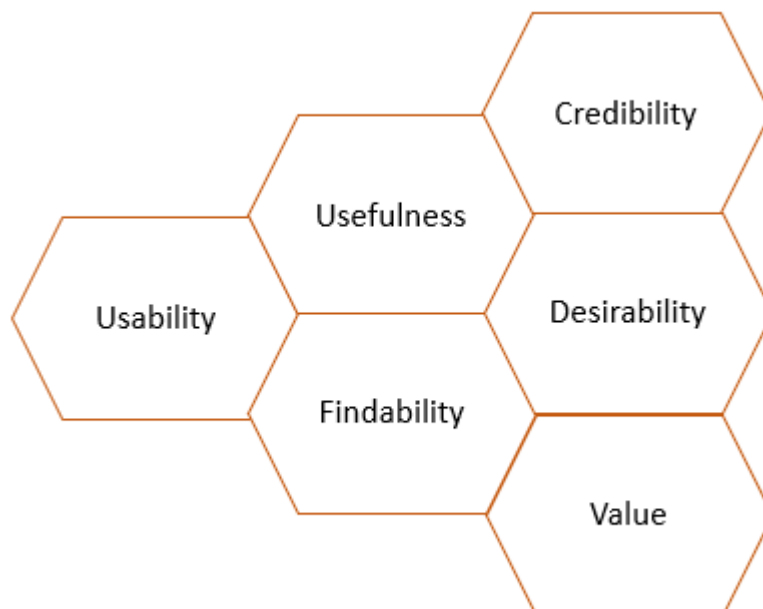
Theorie Usability vs UX

Vorig hoofdstuk hebben we de aspecten van usability behandeld. Daarnaast hebben we het verschil tussen usability en user experience besproken. Op dit laatste gaan we nu dieper in.

Eerder is al verteld dat usability een aspect is van user experience. Hieronder vind je alle aspecten van user experience zoals deze benoemd worden door Peter Morville, een bekende informatie architect. Morville benoemt bij deze aspecten ook accessibility. Accessibility is hieronder niet genoemd omdat het valt onder usability.

De user experience aspecten zijn:

- 1. Usability**
- 2. Usefulness**
De mate waarin het product nut heeft voor de gebruiker. Hierbij worden voordelen als plezier of aantrekkelijkheid ook als nuttig beschouwd.
- 3. Findability**
De mate waarin informatie die het product bevat gemakkelijk gevonden kan worden.
- 4. Credibility**
De mate waarop een gebruiker kan vertrouwen dat het product of de website de beloofde functies uit kan voeren zonder problemen en dit ook blijft doen voor een redelijke tijd.
- 5. Desirability**
De mate waarin het product door uiterlijk, gedrag en service, tegemoetkomt aan vaak onuitgesproken gebruikersverwachtingen.
- 6. Value**
De mate waarin het product toegevoegde waarde levert aan de gebruiker.



Figuur 4: De zes aspecten van user experience.



OPDRACHT 4.

QUIZ

Usability of niet? Je krijgt statements voorgelegd en moeten aangeven of dit statement hoort bij usability of niet.



OPDRACHT 5.

UX ASPECTEN

Kijk nog even naar je antwoord op opdracht 3. Werk je beschrijving verder uit.

Neem de aspecten usefulness, findability, credibility, desirability en value over. En beschrijf onder elk aspect hoe je dat in jouw gekozen website kan terugvinden.

Geef daarna 2 voorbeelden van jouw website waarin je duidelijk het verschil laat zien tussen User Experience (UX) en Usability.

Zorg voor heldere tussenkopjes. Let op dat je het resultaat moet kunnen lezen zonder dit boekje erbij te moeten pakken. Leg dus ook kort in eigen woorden uit wat je aan het beschrijven bent.

Onderstreep de tekst waarvan jij denkt dat je die ook wel in je eigen ontwerp zou willen gebruiken.



OPDRACHT 6.

ZELF REGELS BEDENKEN *

Bedenk zelf drie regels die ervoor kunnen zorgen dat een product of website een goede usability bereikt. Een voorbeeld van een regel is: *Zorg ervoor dat er genoeg ondersteuning is voor de gebruiker.*

1. _____
2. _____
3. _____

Extra

Bespreek en vergelijk je antwoorden met de persoon naast je. Wat zijn de overeenkomsten? En wat de verschillen?



HUISWERK LEESOPDRACHT RICHTLIJNEN

Deze leesopdracht beschrijft een aantal richtlijnen waar je je aan kunt houden voor het bereiken van een goede usability. Lees onderstaande tekst en maak voor jezelf een samenvatting van elke heuristiek. Deze informatie zul je in les 3 nodig hebben.

Jakob Nielsen, een bekende onderzoeker op het gebied van user experience, heeft een aantal richtlijnen geformuleerd die je gebruiken kan tijdens het ontwerpen en optimaliseren van een nieuw product. Deze richtlijnen zijn bedoeld om inzicht te geven in de meest voorkomende problemen van usability en hulp te bieden bij het ontwerpen van een product of systeem.

De heuristieken komen oorspronkelijk uit 1994, dus al een tijd geleden. Toch kun je deze nog goed toepassen. Dit komt omdat het universele vuistregels zijn die makkelijk toe te passen zijn op allerlei producten of services.

Dit zijn de richtlijnen van Nielsen:

1: Laat zien wat er gebeurt (zichtbaarheid van status).

Deze heuristiek verwijst naar hoe goed de status van het systeem aan de gebruiker getoond wordt. Systemen moeten altijd hun gebruikers op de hoogte houden van wat er gebeurt, door middel van geschikte feedback binnen redelijke tijd.



Stel je voor dat je op de lift wacht: je drukt op de knop en je verwacht dat het lampje gaat branden. Op deze manier laat het systeem zien dat je verzoek om de lift te gebruiken is ontvangen, en dat de lift zo zal komen. Misschien zou je zelfs ook de verdieping kunnen zien waar de lift zich bevindt, zodat je kunt inschatten hoe lang je nog moet wachten.

Open communicatie tussen het systeem en de gebruiker helpt de gebruiker niet alleen te begrijpen wat er gebeurt, maar geeft hem of haar ook de macht om beslissingen te maken op basis van deze informatie.

Deze richtlijn geeft aan dat het belangrijk is na te denken over welke informatie de gebruiker moet zien, en hoe dat vlot en helder gepresenteerd kan worden.

Afbeelding 1: Een lift console.

De status van een systeem zou constant getoond kunnen worden, zoals de batterij status of de WiFi indicator, maar zou ook in de vorm van feedback gegeven kunnen worden. Feedback wordt pas zichtbaar nadat de gebruiker een handeling heeft uitgevoerd. Deze feedback is bijvoorbeeld te vinden op websites. Op een site verwachten we dat wanneer we een knopje aanklikken, deze van kleur verandert of dat er een progressiebalkje verschijnt.

Let wel op dat zichtbaarheid van de status niet betekent dat de gebruiker alles moet kunnen zien. Als de gebruiker niks met de gegeven informatie kan, is het ook niet nuttig. Teveel informatie is daarnaast ook afleidend. Focus je als ontwikkelaar dus op de meest belangrijke informatie die behulpzaam is voor de gebruiker. Open communicatie van het systeem maakt het betrouwbaar en voorspelbaar doen aanvoelen. Dit creëert vertrouwen.

2: Overeenkomst systeem en echte wereld

Jakob Nielsen's tweede richtlijn zegt dat het systeem moet overeenkomen met de echte wereld. Dit houdt in dat het systeem woorden, zinnen en concepten moet gebruiken die vertrouwd zijn voor de gebruiker. Een voorbeeld hiervan is de betiteling van een webpagina waar de contact gegevens van werknemers staan. Normaal gesproken heet deze pagina 'contact'. Zo nu en dan gebruiken websites een meer modieuze naam, zoals 'meet & greet'. Deze titel kan verwarrend zijn, omdat het niet per se de contact pagina aanduidt. Een gebruiker zou bijvoorbeeld ook kunnen verwachten dat hier afspraken gemaakt kunnen worden.

Het systeem moet er ook voor zorgen dat informatie in een logische volgorde staat. De werking van het product moet dus overeenkomen met hoe dat in de echte wereld gebeurt. Bijvoorbeeld in een E-Reader kun je de pagina's omslaan door te swipen, net zoals je in een boek de pagina's omslaat.



Afbeelding 2: E-reader en boeken.

3: Gebruiker heeft controle en vrijheid

De derde richtlijn van Nielsen richt zich op de controle en vrijheid van de gebruiker. Het kan bijvoorbeeld gebeuren dat een gebruiker een proces in het systeem per ongeluk start. Hierom moet een gebruiker op elk moment een optie hebben om die ongewilde toestand weer te verlaten of te beëindigen. De functies 'ongedaan maken' en 'opnieuw doen' zijn hier een goed voorbeeld van. Ook de 'terug' en 'verder' knoppen in je browser volgen deze richtlijn. De opties om een proces te verlaten moeten duidelijk aangegeven worden. Zo moet het dus niet zo zijn, dat deze optie alleen gedaan kan worden door bijvoorbeeld je apparaat te schudden. Dit is eerder een functie waar een gebruiker toevallig achter komt dan een functie waar de gebruiker vanaf weet.

Gebruikers krijgen een gevoel van vrijheid, als ze op een link kunnen klikken zonder dat dit onomkeerbare consequenties heeft. Als de link niet blijkt te zijn wat ze verwacht hebben, kunnen ze dit altijd ongedaan maken. Ze zijn altijd in controle over het systeem, en zullen niet per ongeluk vast komen te zitten of acties uitvoeren die zij niet wilden.



Afbeelding 3: De 'terug', 'verder' en 'opnieuw' knoppen in je browser.

4: Wees consistent en volg standaarden

Het is belangrijk voor een systeem om consistentie te behouden en standaarden te volgen. Dit zorgt ervoor dat het systeem voorspelbaar is en gemakkelijk te leren. Gebruikers moeten zich niet afvragen of verschillende woorden, symbolen, situaties of acties op een website iets anders betekenen dan zoals ze gewend zijn.

Consistentie kan zowel intern als extern toegepast zijn. *Interne consistentie* gaat over het behouden van consistentie binnen een product of een set producten. In deze lesmodule zijn bijvoorbeeld alle onderdelen aangegeven door het gebruik van blokken met daarin de titel van het onderdeel. In het geval dat ook de belangrijke termen uit dit hoofdstuk in blokken geschreven waren, zou dit verwarring opleveren omdat het de betekenis van de blokken schendt.

Externe consistentie draait om het behouden van consistentie buiten het ontwerp van een specifiek product. Het winkelwagentje bij webshops is een voorbeeld hiervan. Gebruikers zijn bekend met dit concept, omdat dit gebruikt wordt voor dezelfde functie in elke webwinkel. Gebruikers hebben bij het winkelwagen icoontje verwachtingen bij de functie, maar ook bij de locatie. Zo is dit icoontje (bijna) altijd rechts bovenin de pagina te vinden. Bij het ontwikkelen van jouw eigen website is het erg belangrijk om ook deze consistentie te volgen. Gebruikers spenderen namelijk meer tijd op andere sites dan op die van jou. Als dan bij jouw website die consistentie ontbreekt, forceer je mensen om iets nieuws te leren. Mogelijk is het goed om de consistentie te doorbreken, omdat jij een nieuw en beter patroon hebt bedacht. Máár, je belast de gebruiker elke keer als je hem of haar forceert iets nieuws te leren. Het breken van een externe consistentie moet dus alleen in bijzondere omstandigheden gedaan worden, wanneer het absoluut noodzakelijk is of er heel goede redenen voor zijn.



Afbeelding 4: Het winkelwagentje icoon.

5: Hulp bij fouten

Het voorkomen van gebruikersfouten is kritiek voor een product. Wanneer je naar het ontwerp kijkt, richt je dan eerst op problemen die een grote schade aan zouden kunnen richten. Vanaf daar neem je een stap terug, en kijk je naar dingen die irritant of frustrerend kunnen zijn. Bij een smartphone, bijvoorbeeld, wordt er een foutmelding gegeven wanneer het tijd is om de batterij van de telefoon op te laden. Als de batterij volledig leeg is, kan de telefoon niet meer gebruikt worden. Nu is dit bij een smartphone wel vervelend, maar levert over het algemeen geen grote schade op. Dit is een ander verhaal wanneer een apparaat uitvalt waar mensenlevens afhankelijk van kunnen zijn, zoals bij medische apparaten.

Soms is het voorkomen van fouten makkelijk te doen. Het aanpassen van de grootte, kleur of locatie van een knop kan genoeg zijn. Zelfs het aanpassen van de textuur van een knop kan een mogelijke oplossing zijn. Het laten zien van een bevestigingsbericht kan helpen bij het voorkomen van fouten.

“Weet je zeker dat je geld wilt overmaken naar deze persoon?”. Tot slot helpt ook de ‘ongedaan maken’ knop bij het voorkomen dat fouten onomkeerbaar zijn.



Afbeelding 5: Een bevestigingsbericht na een actie.

6: Zorg dat de bezoeker niets hoeft te onthouden

Deze richtlijn richt zich op het verschil tussen herinneren en herkennen van iets. Herkennen wordt als makkelijker ervaren dan herinneren. Waarom? Als je bijvoorbeeld gevraagd wordt ‘Wat is de hoofdstad van Polen?’, denk je hier langer over na dan wanneer er gevraagd wordt ‘Is Warsaw de hoofdstad van Polen?’. De laatste vraag vereist herkenning. Je kunt namelijk herkennen of de gegeven informatie correct is. Herkenning is gebaseerd op het ophalen van informatie uit je geheugen, gebaseerd op een aantal aanwijzingen.

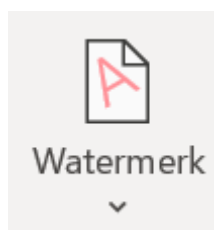
De eerste vraag bevatte twee aanwijzingen:

‘Wat is de hoofdstad van Polen?’

En de tweede vraag bevatte drie aanwijzing:

‘Is Warsaw de hoofdstad van Polen?’

Het stimuleren van herkenning is toe te passen in producten. Een voorbeeld: Stel dat je bezig bent met het schrijven van een verslag. In dit verslag wil je zorgen dat niemand dit kan kopiëren zonder jouw toestemming. Je wilt dus iets identificeerbaars naar jezelf op de achtergrond plaatsen. De term voor deze techniek, genaamd watermerken, komt misschien niet meteen in je op. Wanneer je gaat zoeken in het menu, kun je toch heel snel deze techniek vinden, met behulp van het icoontje of de naam.



Afbeelding 6: De knop voor watermerken in Microsoft Word.

Het toepassen van herkenning in producten maakt ze gebruiksvriendelijker.

7: Flexibiliteit en efficiëntie

De zevende richtlijn van Nielsen richt zich op flexibiliteit en efficiëntie. Veel gebruikte functies die efficiëntie ondersteunen zijn kopiëren en plakken. Deze functies kun je doen door iets te selecteren, op je rechtermuisknop te drukken en de gewenste functie te kiezen, maar ook via je toetsenbord (ctrl + c en ctrl + v). Het kunnen uitvoeren van de functies op verschillende manieren brengt flexibiliteit. Als een systeem een gebruiker de keus geeft om acties uit te voeren op verschillende manieren, kan de gebruiker de snelste of meest comfortabele manier voor hem of haar kiezen.

De eerder genoemde sneltoetsen op het toetsenbord, ctrl + c en ctrl + v, zijn accelerators. Een *accelerator* is een optie of actie die een interactie proces versnelt. Deze zijn bedoeld voor de meer ervaren gebruikers die hun weg met het systeem al goed kennen, en de taak zo snel mogelijk gedaan willen hebben.

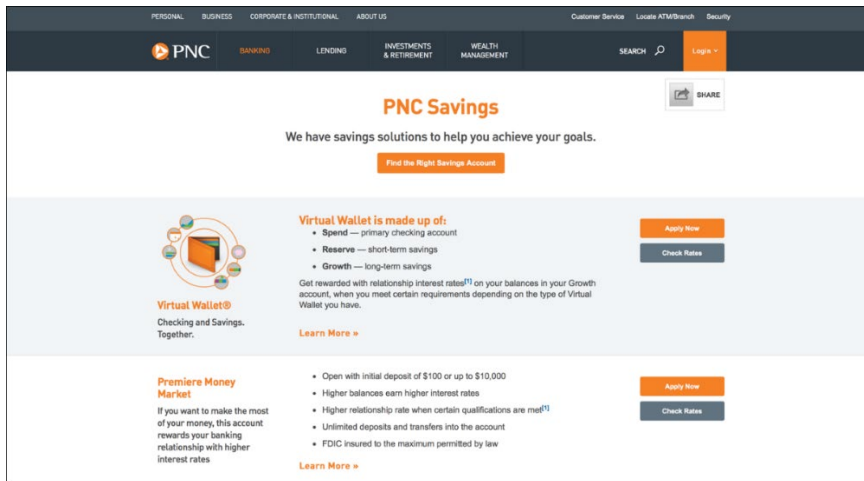
Een systeem moet zowel efficiënt zijn voor experts als gebruiksvriendelijk voor beginners. Beginners moeten in het begin niet elke actie al hoeven te leren. Dit kan namelijk te veel informatie voor hen zijn.

8: Hou het minimaal en verfijnd

Deze richtlijn van Nielsen geeft aan dat het ontwerp minimaal en verfijnd moet zijn. Dit houdt niet in dat je ontwerp plat of met weinig kleur moet zijn. Het gaat erom dat de inhoud en het visuele ontwerp gefocust is op essentiële componenten. Deze richtlijn is gerelateerd aan het mens-machine communicatie concept van de *signaal-ruis verhouding*. Die signaal-ruis verhouding is de verhouding tussen relevante informatie (=signaal) ten opzichte van irrelevante informatie (= ruis). Deze informatie kan zowel tekst inhoud, visuele elementen of zelfs animaties zijn. Alles wat gebruikers moeten verwerken zou kunnen tellen als signaal of als ruis. Om de efficiëntie tijdens het communiceren met de gebruikers via het product te verbeteren is een hoge signaal-ruis verhouding nodig, dus veel belangrijke informatie en weinig ruis. Je kunt dit bereiken door jouw content en kenmerken te prioriteren, zodat je duidelijk is wat het meest relevant is. Informatie die minder vaak gebruikt wordt, of alleen door een kleine groep gebruikers, kun je indelen in irrelevante informatie. Vervolgens zorg je ervoor dat de visuele onderdelen van het product de gebruiker helpen de relevante informatie makkelijk te herkennen.

The screenshot shows the top of the 'THE OLD FARMER'S ALMANAC' website. The header includes a search bar with 'Enter keywords...' and a 'GO' button, followed by the site's name and 'FOUNDED IN 1792'. A navigation menu contains links for WEATHER, ASTRONOMY, GARDENING, CALENDAR, FOOD, ADVICE, and STORE. The main content area is titled 'STORING GLADIOLI CORMS' and includes a sub-header 'In colder regions, dig up gladioli corms once the foliage has faded but before the first fall frost.' Below this is a list of seven bullet points providing detailed instructions on how to dig, dry, store, and replant gladioli corms. At the bottom of the visible section, there is a heading for 'RECOMMENDED VARIETIES'.

Afbeelding 7: Een website met een lage signaal ruis-verhouding. Er is weinig onderscheid tussen relevante en irrelevante informatie. [Bron: NN group]



Afbeelding 8: Een website met hoge signaal-ruis verhouding. Relevante informatie is duidelijk aangegeven door gebruik van kleur en plaatjes. Een gebruiker kan hier makkelijk relevante en irrelevante informatie onderscheiden. [Bron: NN group]

9: Help gebruikers met het erkennen en oplossen van fouten

Het bieden van hulp met het erkennen en oplossen van fouten aan gebruikers kan op verschillende manieren worden bereikt.

Ten eerste is het belangrijk om gebruikers duidelijk te informeren wanneer een fout heeft plaatsgevonden. De simpelste manier om dit te doen is met een foutmelding. Afhankelijk van de context, zou je het bericht kunnen combineren met iets visueels, zoals rode tekst of een waarschuwingsteken. Dit gebeurt bijvoorbeeld wanneer je in een online formulier een verplicht vakje vergeet in te vullen.

Ten tweede moet de gebruiker geïnformeerd worden over wat het probleem is. Hierbij is het gebruik van simpele en duidelijke taal belangrijk.

Tot slot moet er een manier worden aangeboden om de fout te verhelpen. Dit zouden beknopte instructies kunnen zijn, waarmee een gebruiker het probleem kan oplossen, maar het is beter om een makkelijke actie in de foutmelding aan te bieden. Deze makkelijke actie kan bijvoorbeeld bestaan uit een link waar de gebruiker op kan klikken om het probleem op te lossen. Bijvoorbeeld wanneer er geen resultaten zijn voor je zoekopdracht bij Google, vraagt Google of je misschien iets anders bedoelde met daarbij mogelijke zoekopdrachten. Soms kan ook een simpele 'ongedaan maken' knop genoeg zijn.



Afbeelding 9: Een foutmelding met goede informatie.

10: Hulp en documentatie

De laatste richtlijn gaat over hulp en documentatie. Hoewel de meeste producten zijn ontworpen om zo intuïtief mogelijk te zijn, willen gebruikers soms hulp wanneer producten complexer worden. Hulp pagina's zijn nuttig, maar hulp kan ook op andere manieren geïmplementeerd worden. Denk maar aan walkthroughs, tooltips, video's en chatbots.

Bij deze richtlijn moet je je afvragen hoe gemakkelijk het is voor de gebruiker om hulp te zoeken. Ook een uitleg die gericht is op de taak van de gebruiker is belangrijk. Deze documentatie moet concrete stappen aanbieden die uitgevoerd kunnen worden door de gebruiker. Documentatie wordt niet graag gelezen. Een gebruiker moet daarom liefst hulp krijgen op een moment dat hulp nodig zou kunnen zijn, zonder hier specifiek om te hoeven vragen.

Een voorbeeld hiervan is de pop-up die verschijnt in de zoekbalk van Skyscanner. Wanneer je bijvoorbeeld niet zou weten waar je heen kan, biedt Skyscanner de optie aan om dit te bekijken.

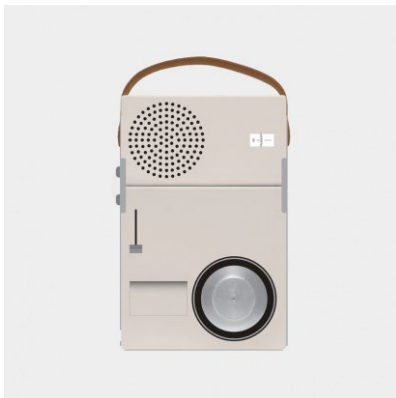


Afbeelding 10: De hulpfunctie bij skyscanner.

LES 2 HEURISTIEKEN

Vorige les hebben we gekeken naar de tien heuristieken van Nielsen. Deze heuristieken zijn richtlijnen die kunnen toegepast worden om de usability van een product te vergroten.

We weten al dat usability een onderdeel is van user experience, waarbij usability zich richt op de gebruiksvriendelijkheid en bruikbaarheid van een product. User experience richt zich verder ook op het ontwerpen van producten met meer focus op het uiterlijk van het product. Dit betekent niet perse dat een product er mooi uit moet zien. Het ontwerp van een product moet passen bij de functie, locatie en gebruikswijze van een product. Design wordt gebruikt om het doel van het product te ondersteunen. Deze vorm van design wordt ook wel product design genoemd en valt onder user experience.



Een bekende naam op het gebied van user experience is Dieter Rams. Dieter Rams is een Duitse meubelmaker, architect en industrieel vormgever. In 1970 werd hij steeds bezorgder over de vormgeving rondom producten. Hij vond dat er verwarring was in formaten, kleuren en geluiden. Daarom vroeg hij zich af 'Is mijn design een goed design?'. Op basis hiervan ontwikkelde hij tien principes. Aangezien product design gaat om het ondersteunen van de functie van het product vallen een aantal van deze principes ook onder usability.

Afbeelding 11: Een design van Dieter Rams

Met deze tien principes wees hij designers op het gebruik van design ter ondersteuning van de functie van hun product:

- 1. Een goed ontwerp maakt het product nuttig**
Een product wordt gekocht om gebruikt te worden. Het moet voldoen aan een aantal criteria, niet alleen functioneel maar ook psychologisch en esthetisch.
- 2. Een goed ontwerp is innovatief**
Technische ontwikkeling geeft altijd nieuwe mogelijkheden voor innovatieve designs. Een design moet altijd mee gaan met deze technische ontwikkelingen.
- 3. Een goed ontwerp is esthetisch**
De esthetische kwaliteit van een product voegt waarde toe omdat producten dagelijks gebruikt worden en dus een effect hebben op de gebruiker en zijn welzijn.
- 4. Een goed ontwerp maakt een product begrijpelijk.**
De functie van een product kan beter worden uitgedrukt wanneer een product voldoet aan de intuïtie van de gebruiker. Een product moet vanzelfsprekend zijn.
- 5. Een goed design is onopvallend**
Producten zijn gereedschappen, geen decoratie of kunst. Het design van een product moet neutraal en teruggehouden zijn zodat de gebruiker de ruimte waarin het product zich bevindt zelf kan inrichten.

6. Een goed design is eerlijk

Het design maakt het product niet meer innovatief, machtig of waardevol dan het eigenlijk is.
Het probeert de gebruiker niet te manipuleren.

7. Een goed design is langdurig

Het design probeert niet modebewust te zijn en kan daarom ook niet uit de mode zijn.

8. Een goed design is grondig ontworpen tot aan het laatste detail.

Niets aan het design moet nagelaten zijn. Zorg en zorgvuldigheid tonen respect voor de gebruiker.

9. Een goed design is milieuvriendelijk

Het gebruik van materialen moet geminimaliseerd worden. Fysieke en visuele vervuiling veroorzaakt door het product tijdens de levenscyclus van het product moeten tot een minimum worden beperkt.

10. Een goed design is zo minimaal mogelijk.

Een design met zo min mogelijk design concentreert zich op de essentiële onderdelen van het product, in plaats van die niet-essentiële onderdelen.

Tijdens de les zal de docent deze principes verder uitleggen.



OPDRACHT 7. ASPECTEN PLAATSEN

Geef in onderstaande tabel bij elk design principe van Dieter Rams de aspecten van user experience aan waaronder het principe valt. De aspecten van user experience zijn: usability, usefulness, desirability, findability, credibility en value. Het kan zijn dat je niet voor al deze aspecten een passend design principe vindt.

Design principe	UX Aspect
1. Een goed ontwerp maakt het product nuttig	
2. Een goed ontwerp is innovatief	
3. Een goed ontwerp is esthetisch	
4. Een goed ontwerp maakt een product begrijpelijk.	
5. Een goed design is onopvallend	
6. Een goed design is eerlijk	
7. Een goed design is langdurig	
8. Een goed design is grondig ontworpen tot aan het laatste detail.	
9. Een goed design is milieu vriendelijk	
10. Een goed design is zo minimaal mogelijk.	

OPDRACHT 8. RICHTLIJNEN PLAATSEN

Product design gaat om het ondersteunen van de functie van het product. Een aantal van de design principes van Dieter Rams vallen onder usability. De principes die onder usability vallen kunnen dan weer onder een heuristiek van Nielsen vallen.

Geef bij de onderstaande design principes van Dieter Rams aan onder welke richtlijn van Nielsen deze zou vallen.

- 4. Een goed ontwerp maakt een product begrijpelijk.
- 6. Een goed design is eerlijk
- 8. Een goed design is grondig ontworpen tot aan het laatste detail.
- 10. Een goed design is zo minimaal mogelijk.

Vul hiervoor onderstaande tabel in.

Design principe	Richtlijn van Nielsen
4. Een goed ontwerp maakt een product begrijpelijk.	
6. Een goed design is eerlijk	
8. Een goed design is grondig ontworpen tot aan het laatste detail.	
10. Een goed design is zo minimaal mogelijk.	



OPDRACHT 9.

VERBETEREN VANUIT HEURISTIEKEN

Deze opdracht maak je in groepen van vier of vijf leerlingen. Elke groep krijgt van de docent twee heuristieken van Nielsen toegewezen. Geef bij deze heuristieken zo concreet mogelijk aan hoe een niet gebruiksvriendelijke website, www.arngren.net veranderd kan worden om alsnog aan deze richtlijnen te voldoen.

Noteer jullie bevindingen hieronder.

Verbetering vanuit eerste heuristiek: _____

Verbetering vanuit tweede heuristiek: _____

Extra

Na de vorige opdracht worden er nieuwe groepen gemaakt, bestaande uit één teamlid van elke vorige groep. In elke groep zit nu één leerling van groep 1, één leerling van groep 2, etc. In deze nieuwe groep bespreekt iedereen de geanalyseerde heuristieken uit zijn vorige groep. Hierbij leg je de antwoorden die je bij bovenstaande opdracht hebt gegeven uit aan de groep.

OPDRACHT 10. ONDERZOEK STARTEN

Maak deze opdracht op maken.mijn-in.nl.

Doel onderzoek

We gaan de website uit de praktische opdracht ook evalueren. Voordat we dat kunnen doen moet je het volgende doen. Beschrijf wat het doel van de evaluatie is. Voorbeelden van zo'n doel zijn:

- Ik wil dat klanten makkelijker een artikel kunnen aanschaffen op mijn webwinkel dan nu het geval is.
- Ik denk dat het tekstadvies op deze website niet geschikt is voor de doelgroep.
- Ik wil graag weten of de afbeeldingen op deze informatiepagina's duidelijk zijn voor de eindgebruiker.
- Ik wil weten of de nieuwe functionaliteit in de applicatie aan de verwachtingen voldoet.

Onderzoeksvraag

Stel ook een onderzoeksvraag op.

Bedenk een onderzoeksvraag die je tijdens je opdracht wilt beantwoorden. Gebruik daarbij de volgende hulpvragen:

- Wat wil je precies onderzoeken of te weten komen?
- Wat valt je op aan het product of systeem dat je onderzoekt?
- Wat verwacht je dat gebruikers lastig of juist makkelijk zullen vinden?
- Kun je je onderzoeksvraag beginnen met "In hoeverre...", "Hoe...", of "Wat is het effect van..."?

Voorbeeld: "In hoeverre vinden gebruikers de navigatie op de homepage van [naam website] duidelijk?"

Hypothese

Geef het verwachte antwoord op je onderzoeksvraag.

LES 3 INTRODUCTIE HTML

HTML is de afkorting van **HyperText Markup Language**. Het is de basis van bijna elke webpagina op het internet. Als je ooit een website hebt bezocht, zoals Wikipedia of YouTube, dan heb je eigenlijk een HTML-pagina bekeken. HTML bepaalt de **structuur** van een webpagina: wat waar staat, of iets een titel of een paragraaf is, of er een afbeelding getoond wordt, enzovoort. Je kunt HTML zien als de ruggengraat van een website.

Wat doet HTML?

HTML vertelt een webbrowser – zoals Chrome, Firefox of Safari – hoe een pagina eruit moet zien. Het beschrijft **inhoud** (zoals tekst, plaatjes en knoppen) en hoe die inhoud is opgebouwd. HTML maakt bijvoorbeeld duidelijk wat de hoofdtitel van de pagina is, waar een lijstje staat, of waar een link naartoe verwijst.

HTML bestaat uit **elementen** die worden opgeschreven in **tags**. Een tag is een soort code tussen haken, zoals `

` voor een paragraaf of `

` voor een hoofdtitel. De meeste HTML-elementen bestaan uit een **openende tag**, de **inhoud**, en een **sluitende tag**. Bijvoorbeeld:

```
<p>Dit is een stukje tekst.</p>
```

Hier geeft `

` aan dat het om een paragraaf gaat, en `

` sluit die paragraaf af.

Hoe ziet een HTML-pagina eruit?

Een eenvoudige HTML-pagina begint met een vaste structuur. Hieronder zie je een voorbeeld:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mijn eerste pagina</title>
  </head>
  <body>
    <h1>Welkom!</h1>
    <p>Dit is mijn eerste webpagina.</p>
  </body>
</html>
```

- `

` vertelt dat dit een moderne HTML-pagina is.
- `` is het begin van de pagina.
- Binnen `` staan gegevens die de pagina beschrijven (zoals de titel).
- In `` komt alles wat je op het scherm ziet, zoals tekst en afbeeldingen.

Waarom is HTML belangrijk?

Als je leert programmeren voor het web, begin je bijna altijd met HTML. Het is niet moeilijk om te leren, en je ziet direct resultaat: je maakt iets dat zichtbaar is in de browser. Bovendien werkt HTML samen met andere technologieën, zoals **CSS** (voor opmaak en kleuren) en **JavaScript** (voor interactie en gedrag). Samen vormen ze de bouwstenen van het internet.

OPDRACHT 11. JE EERSTE WEBPAGINA

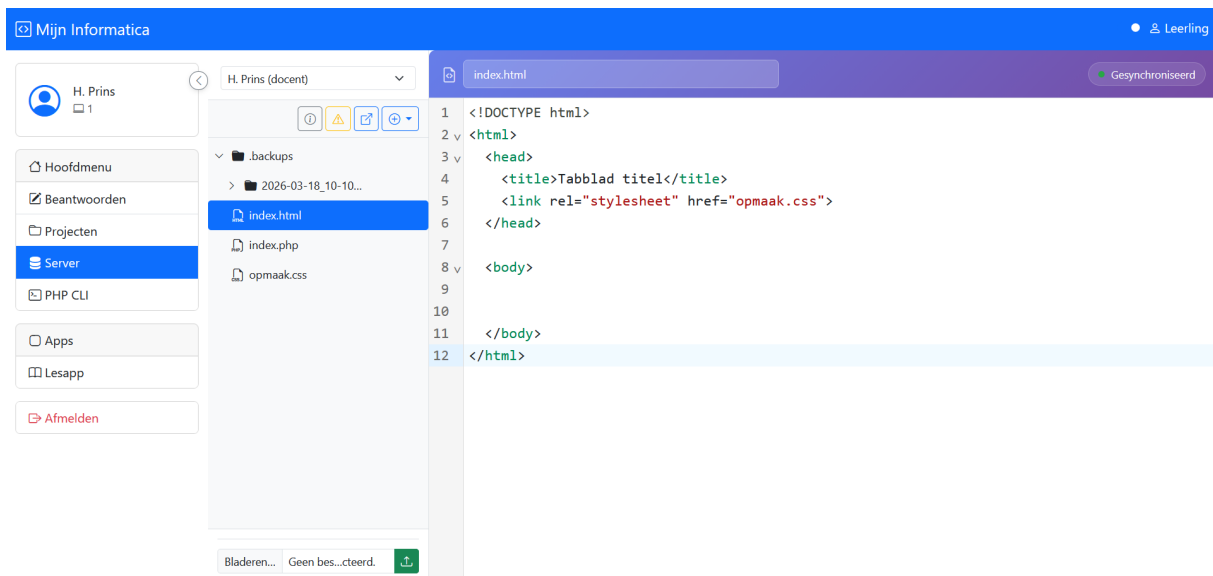
Je maakt je allereerste HTML-pagina die de tekst "Hello World" laat zien in een webbrowser. Je leert hoe HTML-bestanden eruitzien en hoe je ze kunt bekijken.

Log in op maken.mijn-in.nl met je Google account van school

Server

In maken.mijn-in.nl kun je kiezen voor server. Daarin zie je twee vensters:

- Links: je bestanden (zoals index.html)
- Rechts: waar je je code typt



Maak of bewerk index.html en plaats de code daar met dit voorbeeld:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World Pagina</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>Dit is mijn eerste webpagina.</p>
  </body>
</html>
```

Kies het 3^e icoon op het resultaat te bekijken. Je webpagina verschijnt in nieuw tabblad.

Je zou nu groot moeten zien staan:

Hello World!

En daaronder een zin:

Dit is mijn eerste webpagina.

UITLEG: DE BELANGRIJKSTE HTML-TAGS

HTML werkt met **tags**. Een tag vertelt de browser wat voor soort inhoud iets is: een kop, een paragraaf, een link, enzovoort. De meeste tags bestaan uit een **openende tag** en een **sluitende tag**. Bijvoorbeeld:

```
<p>Dit is een paragraaf.</p>
```

Hier is `<p>` de **openende tag**, `</p>` de **sluitende tag**, en alles ertussen is de inhoud.

`<html>...</html>`

Dit is de **buitenkant** van je hele HTML-document. Alles komt **binnen deze tag** te staan.

```
<html>
  <!-- Hier komt je hele pagina -->
</html>
```

`<!DOCTYPE html>`

Deze regel komt **bovenaan** elk HTML-bestand. Het zegt tegen de browser: "Dit is een moderne HTML5-pagina". Dit is geen echte tag, maar wel verplicht.

```
<!DOCTYPE html>
```

`<head>...</head>`

Hierin zet je **informatie over de pagina** die niet direct zichtbaar is. Bijvoorbeeld de titel van de pagina of koppelingen naar opmaak (CSS).

```
<head>
  <title>Mijn website</title>
</head>
```

`<title>...</title>`

Deze tag staat **in de <head>**. De tekst hierin zie je bovenin het tabblad van de browser.

```
<title>Welkom op mijn site</title>
```

`<body>...</body>`

Alles wat **zichtbaar is op je pagina** komt in de `<body>`: koppen, tekst, afbeeldingen, enzovoort.

```
<body>
  <h1>Welkom!</h1>
  <p>Leuk dat je er bent.</p>
</body>
```

`<h1>...</h1>` t/m `<h6>...</h6>`

Dit zijn **koppen** (headings). `<h1>` is de **belangrijkste en grootste** kop, `<h6>` de kleinste.

```
<h1>Hoofdtitel</h1>
<h2>Subtitel</h2>
```

Gebruik **slechts één <h1>** per pagina, voor de hoofdonderwerp.

`<p>...</p>`

Gebruik je voor een **paragraaf**, dus gewone tekst.

```
<p>Dit is een stukje uitleg.</p>
```

`<a href="...">...</a>`

Dit is een **hyperlink**. Met href geef je aan waar de link naartoe gaat.

```
<a href="https://www.google.com">Ga naar Google</a>
```

``

Toont een **afbeelding**. src is het adres van de afbeelding, alt is de beschrijving (handig voor als de afbeelding niet laadt).

```

```

Let op: deze tag **sluit je niet af** met `</img>`. Hij staat op zichzelf.

`<ul>`, `<ol>` en `<li>`

Voor **lijstjes**:

- `<ul>` = unordered list = **lijst met bolletjes**
- `<ol>` = ordered list = **genummerde lijst**
- `<li>` = list item = **één onderdeel van de lijst**

```
<ul>  
  <li>Appel</li>  
  <li>Banaan</li>  
</ul>
```

Samenvatting

Tag	Wat doet het?
<code><html></code>	Omvat de hele pagina
<code><head></code>	Informatie over de pagina (niet zichtbaar)
<code><title></code>	Tekst in het browsertabblad
<code><body></code>	Alles wat zichtbaar is op de pagina
<code><h1></code> t/m <code><h6></code>	Koppen, van groot naar klein
<code><p></code>	Paragraaf (tekst)
<code><a></code>	Link naar een andere pagina
<code></code>	Afbeelding tonen
<code></code> , <code></code> , <code></code>	Ongeordende of genummerde lijst

Als je deze tags kent en gebruikt, kun je al een hoop maken met HTML! In de volgende stap kun je gaan oefenen met echte voorbeelden.

OPDRACHT 12. MAAK JE EIGEN PROFIELPAGINA

Je maakt een eenvoudige **persoonlijke webpagina** met HTML. Je oefent met koppen, paragrafen, lijsten, links en een afbeelding. Let op dit doen we niet op de server, maar bij de opdrachten op mijn-in.nl. Hier staat bij een aantal opdrachten ook al voorbeeldcode ingevuld die je kan gebruiken.

Wat moet je maken?

Een webpagina over jezelf met:

1. Een **hoofdtitel** met je naam (<h1>)
2. Eén of meer **paragrafen** waarin je iets over jezelf vertelt (<p>)
3. Een **genummerde lijst** van je drie favoriete vakken ()
4. Een **lijst met hobby's** met bolletjes ()
5. Een **afbeelding** van iets dat bij jou past ()
6. Een **link** naar een website die je vaak bezoekt (<a>)

Extra uitdaging

- Voeg extra kopjes toe, bijvoorbeeld "Mijn favoriete eten".
- Zet de afbeelding op een andere plek in de pagina.
- Gebruik <h2> en <h3> voor structuur.

Uitleg: Wat zijn blokken in HTML

In HTML wordt de inhoud van een webpagina opgedeeld in **blokken**. Dit helpt om overzicht te houden én om een pagina goed te kunnen indelen.

Een **blok** is een stuk inhoud dat vanzelf op een **nieuwe regel** begint en vaak een eigen ruimte op de pagina krijgt. Denk aan een paragraaf, een kop, een afbeelding of een hele sectie van de pagina. Deze blokken worden ook wel **block-level elementen** genoemd.

Er bestaan twee soorten HTML-elementen als het gaat om opbouw:

- **Blokelementen (block-level)**
- **Inline-elementen**

Blokelementen

Een **blokelement** vult de **hele breedte** van de pagina (of van de container waar het in zit), zelfs als de inhoud kort is. Het begint altijd op een **nieuwe regel**.

Voorbeelden:

- <div> – een leeg blok dat je kunt gebruiken om inhoud te groeperen
- <p> – een paragraaf
- <h1> t/m <h6> – koppen
- , , – lijsten
- <section>, <header>, <footer>, <article> – onderdelen van de layout

Voorbeeld:

```
<h1>Welkom</h1>
<p>Dit is een blok tekst.</p>
<p>En dit is nog een paragraaf.</p>
```

Elke paragraaf komt **onder** de vorige, omdat <p> een blokelement is.

Inline-elementen

Een **inline-element** komt **in de regel** te staan waar je al bezig bent. Het breekt de regel **niet af**. Deze gebruik je voor kleine stukjes opmaak **binnen** een blok.

Voorbeelden:

- <a> – link
- – vetgedrukte tekst
- – schuingedrukte tekst
- – een klein stukje dat je apart kunt opmaken

Voorbeeld:

```
<p>Dit is <strong>belangrijke</strong> tekst in een
paragraaf.</p>
```

Hier zie je dat de -tag **binnen** de <p>-tag wordt gebruikt. Het is inline, dus de tekst blijft op dezelfde regel.

Waarom zijn blokken handig?

Met blokken kun je:

- Je pagina logisch indelen
- Inhoud groeperen
- De opmaak overzichtelijk houden (voor jezelf én voor CSS later)
- Duidelijk aangeven waar onderdelen beginnen en eindigen

Tip: Gebruik <div> om blokken te groeperen

Een <div> is een **lege container** die je kunt gebruiken om andere elementen bij elkaar te zetten.

Handig als je later delen wilt opmaken of verplaatsen.

```
<div>
  <h2>Over mij</h2>
  <p>Ik ben leerling in 4 vwo.</p>
</div>
```

Samenvatting

Type element	Gedrag	Voorbeelden
Blokelement	Begint op nieuwe regel, vult breedte	<p>, <div>, <h1>
Inline-element	Blijft in de regel staan	<a>, ,



OPDRACHT 13.

WERKEN MET BLOKKEN EN INLINE ELEMENTEN

Je oefent het verschil tussen blokken en inline elementen door een eigen HTML-pagina te maken waarin je beide soorten gebruikt.

Opdracht

Maak een HTML-pagina met de volgende onderdelen:

1. Een **hoofdtitel** met jouw naam (<h1>)
2. Een **blok** over jezelf (<div>) met daarin:
 - a. Een **subtitel**: "Over mij" (<h2>)
 - b. Een **paragraaf** over jezelf met minstens twee **inline elementen**:
 - i. een stuk **vetgedrukte tekst** ()
 - ii. een stuk **schuingedrukte tekst** ()
3. Een tweede blok (<div>) met:
 - a. Een **subtitel**: "Mijn interesses" (<h2>)
 - b. Een lijstje van 3 hobby's met een ongeordende lijst ()
4. Onderaan een zin met een **link** (<a>) naar je favoriete website, verwerkt in een **paragraaf**

Extra uitdaging

- Voeg een <hr> toe tussen de blokken voor een visuele scheiding
- Zet een extra rond een woord in je tekst en geef er in CSS een kleur aan (optioneel)

UITLEG: WAT ZIJN CONTAINERS IN HTML

In HTML gebruik je vaak **containers** om onderdelen van een pagina te groeperen. Een **container** is een element dat je gebruikt om meerdere andere elementen samen te voegen in één blok. Dat helpt je om de pagina logisch in te delen en later makkelijker op te maken met CSS.

De bekendste container is de <div>-tag.

Waarom gebruik je containers?

1. **Structuur**: Je kunt je pagina opdelen in duidelijke stukken, zoals "Over mij", "Interesses", of "Contact".
2. **Groepering**: Je plaatst bij elkaar horende inhoud samen — bijvoorbeeld een titel en een paragraaf.
3. **Opmaak**: Je kunt later een container een achtergrondkleur, rand of positie geven met CSS.
4. **Herbruikbaarheid**: In frameworks (zoals Bootstrap) worden containers gebruikt om standaardblokken te maken.

De <div>-tag

De <div> is een **lege, neutrale container**. Hij zegt zelf niets over de inhoud, maar wordt gebruikt om andere elementen in te stoppen.

Voorbeeld:

```
<div>
  <h2>Over mij</h2>
  <p>Ik zit in 4 havo en hou van tekenen.</p>
</div>
```

Hiermee geef je aan: "Deze kop en paragraaf horen bij elkaar".

Andere soorten containers

Naast <div> bestaan er ook **semantische containers**, die wél iets zeggen over de inhoud. Dit helpt bij leesbaarheid (voor mensen én zoekmachines).

Voorbeelden:

- <header> – bovenste gedeelte van de pagina
- <nav> – navigatiebalk
- <main> – hoofdinhoud
- <section> – een logische sectie van de inhoud
- <article> – een losstaand stuk, bijvoorbeeld een blogbericht
- <footer> – onderste gedeelte van de pagina

Voor beginners is <div> meestal voldoende, maar het is goed om te weten dat er dus ook "betekenisvolle containers" bestaan.

Samenvatting

Container	Wat het doet
<div>	Algemene container voor inhoud
<section>	Logisch onderdeel van een pagina
<article>	Zelfstandig stuk inhoud
<header>	Bovenkant van de pagina of een blok
<footer>	Onderkant van de pagina of een blok
<nav>	Navigatie, zoals menu's
<main>	Belangrijkste inhoud van de pagina



OPDRACHT 14. WEBSITE MET STRUCTUUR

Je maakt een webpagina die is ingedeeld in duidelijke onderdelen: een kop (header), een menu (nav), de hoofdinhoud (main) en een onderkant (footer). Je leert hoe je deze **semantische containers** gebruikt.

Wat moet je maken?

Een webpagina over een zelfgekozen onderwerp, bijvoorbeeld "Mijn hobby's", "Mijn favoriete sport", of "Mijn schoolweek", met deze structuur:

1. **Header** met een titel van de pagina
2. **Navigatie** met drie links (de links hoeven niet echt te werken)
3. **Main**-gedeelte met minstens twee paragrafen
4. **Footer** met je naam en het jaartal

Extra uitdagingen

- Voeg een afbeelding toe in de <main>
- Gebruik <section> binnen <main> om paragrafen te groeperen

UITLEG: WAT ZIJN HTML-CONVENTIES

Conventions zijn **afspraken of richtlijnen** over hoe je HTML schrijft. Ze zijn **niet noodzakelijk om het te laten werken**, maar helpen om je code:

- **leesbaar**
- **duidelijk**
- **samenwerkbaar**
te maken — voor jezelf én voor anderen.

Ook professionele webontwikkelaars gebruiken deze conventies om hun HTML netjes en consistent te houden.

Belangrijkste HTML-conventions

1. Gebruik kleine letters voor tags en attributen

HTML is niet hoofdlettergevoelig, maar de standaard is: **alles in kleine letters**.

✓ Goed:

```
<p>Hallo!</p>

```

✗ Niet netjes:

```
<P>Hallo!</P>
<IMG SRC="foto.jpg" ALT="Beschrijving">
```

2. Sluit altijd je tags correct af

De meeste elementen bestaan uit een **openende tag** en een **sluitende tag**. Vergeet de sluitende tag niet.

✓ Goed:

```
<p>Dit is een paragraaf.</p>
```

✗ Fout:

```
<p>Dit is een paragraaf.
```

Uitzondering: sommige elementen, zoals , hebben geen inhoud en hoeven alleen **één enkele tag**.

3. Gebruik inspringing (indentatie)

Als je HTML-elementen **in elkaar nestelt** (in een ander element zet), laat dit dan zien door de binnenste elementen in te springen. Meestal gebruik je **2 of 4 spaties**.

✓ Goed:

```
<body>
  <h1>Titel</h1>
  <p>Tekst</p>
</body>
```

✗ Slecht leesbaar:

```
<body>
<h1>Titel</h1>
<p>Tekst</p>
</body>
```

4. Gebruik sprekende namen voor bestandsnamen en klassen

Als je bestanden, IDs of CSS-klassen een naam geeft, kies dan iets dat **duidelijk en logisch** is. Gebruik liever geen hoofdletters of spaties.

✓ Goed:

- mijn-pagina.html
- class="menu-knop"

✗ Minder goed:

- Page1.HTML
- class="knopje1"

5. Gebruik dubbele aanhalingstekens rond attributen

HTML accepteert enkele of dubbele aanhalingstekens, maar het is conventie om **dubbele aanhalingstekens** te gebruiken.

✓ Goed:

```
<a href="https://example.com">Klik hier</a>
```

✗ Minder netjes:

```
<a href='https://example.com'>Klik hier</a>
```

Waarom zijn conventions belangrijk?

- Ze maken je code **duidelijk** voor anderen (en voor jezelf als je later terugkijkt).
- Ze voorkomen fouten.
- Ze zorgen ervoor dat je makkelijk kunt samenwerken in een team of project.
- Ze zijn een goede voorbereiding op professioneel werken met HTML, CSS en JavaScript.

Samenvatting van conventions

Regel	Waarom?
Kleine letters voor tags	Duidelijk en consistent
Tags goed afsluiten	Voorkomt fouten
Netjes inspringen (indenteren)	Beter leesbaar
Duidelijke namen gebruiken	Begrijpelijk voor iedereen
Dubbele aanhalingstekens	Standaard en duidelijk

OPDRACHT 15. CONVENTIES

Leerlingen oefenen met het herkennen van veelgemaakte fouten en slordigheden in HTML-code en verbeteren deze volgens de conventies.

Bekijk de onderstaande HTML-code. Er staan **vijf dingen fout** of slordig volgens de HTML-conventions.

Schrijf op wat er niet klopt en verbeter de code.

```
<BODY>
<h1> Welkom op Mijn Site </H1>
<p>Dit is mijn eerste pagina
<img SRC='foto.JPG' ALT='Een Foto'>
<a HREF="pagina2.html">Volgende</a>
</body>
```

LES 4 INTRODUCTIE CSS

Je weet nu hoe je met HTML een webpagina maakt. Maar... misschien is je pagina nog een beetje saai. Alles is zwart-wit, standaard lettertypes, geen kleur of indeling. Daar komt **CSS** bij kijken.

Wat is CSS?

CSS staat voor **Cascading Style Sheets**.

Het is de taal waarmee je de **opmaak en vormgeving** van een webpagina bepaalt. Met CSS kun je bijvoorbeeld:

- De **kleur** van tekst en achtergronden veranderen
- Het **lettertype** aanpassen
- Elementen **verschuiven** of naast elkaar zetten
- **Afstanden** en **marges** instellen
- Je website laten werken op **verschillende schermen**

HTML bepaalt **wat** er op de pagina staat.

CSS bepaalt **hoe** het eruitziet.

Hoe schrijf je CSS-regels? – De basisnotatie

CSS bestaat uit **regels** die allemaal dezelfde structuur hebben:

```
selector {  
    eigenschap: waarde;  
}
```

Uitleg:

- **Selector:** zegt *op welk element* de stijl toegepast moet worden (bijvoorbeeld p, h1 of .belangrijk)
- **Accolades {}:** bevatten de eigenschappen die je wilt toepassen
- **Eigenschap:** wat je wilt veranderen (zoals color, font-size, background-color, enz.)
- **Waarde:** de nieuwe waarde die je instelt (zoals red, 20px, center, enz.)
- Vergeet niet: elke regel **eindigt met een puntkomma ;**

Hoe voeg je CSS toe aan HTML?

Er zijn drie manieren om CSS toe te voegen:

1. Inline CSS – direct in een HTML-element

Gebruik het style-attribuut:

```
<p style="color: blue;">Deze tekst is blauw.</p>
```

Simpel, maar alleen geschikt voor kleine testjes.

2. Interne CSS – in het <head>-gedeelte

Schrijf CSS in een <style>-blok in je HTML-bestand:

```
<head>
  <style>
    h1 {
      color: red;
      font-size: 36px;
    }
    p {
      color: gray;
    }
  </style>
</head>
```

Handig voor kleine projecten.

3. Externe CSS – in een apart bestand

Je schrijft de CSS in een los .css-bestand en koppelt dit aan je HTML-bestand:

```
<link rel="stylesheet" href="stijl.css">
```

In stijl.css zet je dan bijvoorbeeld:

```
body {
  background-color: lightyellow;
}

h1 {
  color: darkgreen;
}
```

De beste aanpak voor grotere of meerdere pagina's en de standaard waarmee wij gaan werken.

Voorbeeld: HTML + CSS samen

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: #f0f0f0;
      }
      h1 {
        color: navy;
        text-align: center;
      }
      p {
        font-family: Arial, sans-serif;
        color: #333;
      }
    </style>
  </head>
  <body>
```

```

    <h1>Welkom!</h1>
    <p>Dit is mijn eerste pagina met opmaak.</p>
  </body>
</html>

```

Wat kun je allemaal stylen?

CSS-eigenschap	Wat het doet
color	Kleur van tekst
background-color	Achtergrondkleur
font-size	Grootte van de tekst
font-family	Lettertype
text-align	Uitlijning van tekst (bv. center)
margin, padding	Ruimte rond elementen
border	Rand om een element

Samenvatting

- **HTML = inhoud, CSS = opmaak**
- CSS kan inline, in <style> of in een apart bestand
- Je gebruikt CSS om kleuren, vormen, posities en lettertypes te regelen
- CSS maakt je pagina aantrekkelijk en professioneel

OPDRACHT 16. GEEF JE PAGINA KLEUR EN STIJL

Je oefent met het aanpassen van de **kleur**, **lettertypes** en **uitlijning** van je webpagina met **interne CSS** in een <style>-blok.

Opdracht

Maak een eenvoudige webpagina over jezelf met de volgende eisen:

1. Een **achtergrondkleur** voor de hele pagina
2. Een hoofdtitel (<h1>) die:
 - a. een **aparte kleur** heeft
 - b. **in het midden staat**
3. Eén of meer paragrafen (<p>) met:
 - a. een **ander lettertype**
 - b. een **donkere kleur** tekst
4. Een afbeelding () die **een rand** krijgt met CSS

Extra uitdaging

- Probeer de breedte van de afbeelding aan te passen met width: 50%;
- Voeg een tweede paragraaf toe met een andere kleur (gebruik een extra CSS-klasse)

UITLEG: WAT IS CLASS IN HTML EN CSS?

De **class** is een **naam die je aan een HTML-element geeft** zodat je dat element (of meerdere tegelijk) met **CSS** kunt opmaken. Je kunt het zien als een soort **label**.

Met een class kun je:

- Bepalen **welke CSS-opmaak** bij een element hoort
- **Meerdere elementen tegelijk** dezelfde stijl geven
- Verschillende stijlen toepassen op vergelijkbare elementen

Hoe gebruik je class?

In HTML:

Je voegt een class toe aan een element met het class-attribuut:

```
<p class="belangrijk">Let goed op deze tekst!</p>
```

Dit betekent: “Deze paragraaf heeft de class 'belangrijk'.”

In CSS:

In je <style>-blok of CSS-bestand gebruik je een **punt (.)** vóór de classnaam:

```
.belangrijk {  
  color: red;  
  font-weight: bold;  
}
```

Deze code zegt: “Alle elementen met de class 'belangrijk' krijgen rode, vette tekst.”

```
<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
      .belangrijk {  
        color: red;  
        font-weight: bold;  
      }  
  
      .mijn-kader {  
        border: 2px solid black;  
        padding: 10px;  
        background-color: #f9f9f9;  
      }  
    </style>  
  </head>  
  <body>  
    <p class="belangrijk">Let op: dit is belangrijke  
informatie!</p>  
  
    <div class="mijn-kader">  
      <h2>Over mij</h2>
```

```
<p>Ik zit in 4 havo en houd van muziek en gamen.</p>
</div>
</body>
</html>
```

Meerdere elementen, één class

Je mag dezelfde class aan **meerdere elementen** geven. Dan hebben ze allemaal dezelfde stijl.

```
<p class="belangrijk">Eerste waarschuwing</p>
<p class="belangrijk">Tweede waarschuwing</p>
```

Beide paragrafen krijgen dezelfde CSS-opmaak.

Eén element, meerdere classes

Je kunt ook **meerdere classes tegelijk** gebruiken, gescheiden door een spatie:

```
<p class="belangrijk klein">Let op!</p>
```

Dan kun je in je CSS dus meerdere stijlen combineren:

```
.belangrijk {
  color: red;
}
.klein {
  font-size: 12px;
}
```

Waarom is class belangrijk?

- Je kunt **herbruikbare stijlen** maken (je hoeft niet alles 10x te schrijven)
- Het maakt je code overzichtelijk
- Je kunt later met JavaScript ook werken met classes

Samenvatting

Term	Uitleg
class in HTML	Een naam geven aan een element voor styling
.classnaam in CSS	Zo geef je op hoe die class eruit moet zien
Meerdere elementen	Kunnen dezelfde class delen
Meerdere classes per element	Kan met spaties in één class="" attribuut

OPDRACHT 17. WERKEN MET CLASSES

Je leert hoe je **meerdere elementen een class geeft** en hoe je met CSS bepaalt **hoe die class eruitziet**.

Opdracht

Maak een HTML-pagina over je **favoriete dieren**. Je gebruikt daarbij verschillende classes om bepaalde stukken tekst of afbeeldingen **op te maken met kleur, lettertype en rand**.

Je moet minstens deze drie classes gebruiken:

Classnaam	Wat het moet doen
titel	Grote, gekleurde kop
belangrijk	Vetgedrukte, opvallende tekst
plaatje	Afbeelding met rand en vaste breedte

Extra uitdaging

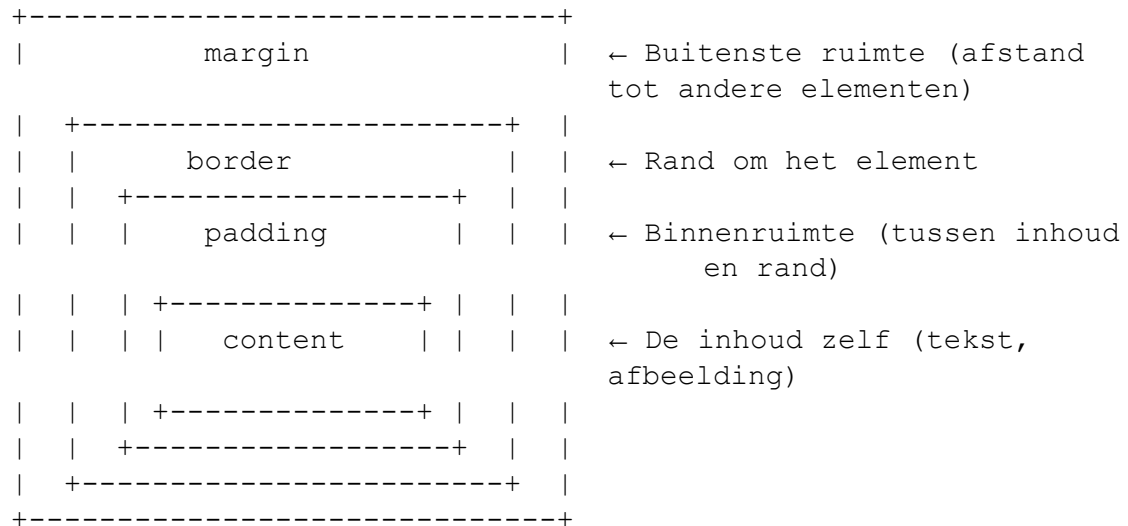
- Maak een extra class zoals mijn-achtergrond om de achtergrondkleur van de pagina te veranderen.
- Geef elke afbeelding een **andere class**, bijvoorbeeld dier1, dier2, en pas ze elk anders aan in CSS.

UITLEG: WAT IS HET CSS BOX MODEL?

Wanneer je een element op een webpagina opmaakt met CSS (zoals een <p>, <div> of), behandelt de browser dat element als een **doos** — een **box**.

Dat noemen we het **box model**.

Het box model bestaat uit **vier lagen** rond elk element:



De vier onderdelen uitgelegd

1. Content

De **inhoud** van het element: bijvoorbeeld tekst of een afbeelding.

2. Padding

De **binnenruimte** tussen de inhoud en de rand. Zorgt dat de tekst of afbeelding niet vastplakt aan de rand.

```
padding: 10px;
```

3. Border

De **rand** om het element. Je kunt dikte, kleur en stijl instellen.

```
border: 2px solid black;
```

4. Margin

De **buitenruimte** rond het element. Zorgt voor afstand tussen dit element en andere elementen.

```
margin: 20px;
```

Voorbeeld in HTML + CSS

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .mijn-doos {
        background-color: lightblue;
```

```

padding: 20px;
border: 3px solid navy;
margin: 30px;
}
</style>
</head>
<body>
  <div class="mijn-doos">
    <p>Dit is een doos met padding, border en margin.</p>
  </div>
</body>
</html>

```

Wat zie je:

- **Lichtblauw vlak** (de div) met tekst erin
- **Ruimte** rond de tekst: dat is de **padding**
- **Donkere rand**: dat is de **border**
- **Witruimte** tussen deze doos en andere dingen op de pagina: dat is de **margin**

Een ezelsbruggetje

Denk aan een **pakketje**:

- De **inhoud** is je cadeau (tekst/afbeelding)
- De **padding** is het inpakpapier rond het cadeau
- De **border** is het plakband of de doos zelf
- De **margin** is de lege ruimte in de verzenddoos of tussen meerdere pakketjes

Waarom is het box model belangrijk?

- Het bepaalt hoe groot elementen **echt** zijn
- Je kunt er mee **indelen en uitlijnen**
- Zonder het box model kun je moeilijk professionele layouts maken



OPDRACHT 18. BOUW JE EIGEN GEKLEURDE DOOS

Je maakt een HTML-element met CSS-opmaak volgens het **box model**. Je oefent met het instellen van **padding**, **margin** en **border**, en ziet wat er verandert.

Opdracht

Maak een webpagina met een <div> die eruitziet als een doos met:

1. **Een achtergrondkleur**
2. **Padding** van minimaal 20 pixels
3. **Een zichtbare border** (dikte, kleur en stijl)
4. **Margin** zodat de doos niet tegen de rand van de pagina zit
5. **Tekst** in de doos die duidelijk niet tegen de rand geplakt zit

Extra uitdaging

- Maak **twee dozen** met verschillende margin- en padding-instellingen
- Zet elke doos een andere **border-style** (dotted, solid, double, etc.)
- Zet een **plaatje** of een lijstje in een van de dozen



OPDRACHT 19. TUSSENTOETSJE BASISKENNIS HTML & CSS

1. Waarvoor staat de afkorting HTML?

- A. High-Tech Markup Language
- B. HyperText Modeling Language
- C. HyperText Markup Language
- D. Home Tool Markup Language

2. Welke tag gebruik je voor een paragraaf in HTML?

- A. <p>
- B. <text>
- C. <paragraph>
- D. <para>

3. Wat doet de <title>-tag?

- A. Laat een knop zien
- B. Geeft een koptekst op de pagina
- C. Zet een afbeelding op het scherm
- D. Toont tekst in het browsertabblad

4. Hoe voeg je een achtergrondkleur toe aan een element in CSS?

- A. background-color: red;
- B. color-background: red;
- C. bgcolor = red
- D. style="red"

5. Wat is de juiste syntax voor een CSS-regel?

- A. selector: eigenschap = waarde
- B. { selector: eigenschap: waarde; }
- C. selector { eigenschap: waarde; }
- D. selector => eigenschap: waarde

6. Wat is het verschil tussen een block-element en een inline-element?

- A. Block-elementen zijn alleen zichtbaar in mobiele weergave
- B. Inline-elementen starten altijd op een nieuwe regel
- C. Block-elementen vullen de hele breedte; inline-elementen niet
- D. Er is geen verschil

7. Welke rol speelt de <div>-tag in HTML?

- A. Het maakt een lijst met opsommingstekens
- B. Het is een container zonder betekenis om inhoud te groeperen
- C. Het zorgt voor opmaak van tekst
- D. Het maakt een navigatiebalk

8. Wat gebeurt er als je padding: 20px; toevoegt aan een element?

- A. De tekst wordt kleiner
- B. Het element krijgt een rand van 20 pixels
- C. De afstand tussen elementen wordt groter
- D. Er komt ruimte tussen de inhoud en de rand van het element

9. Waarom is het gebruik van class handig in HTML?

- A. Je kunt een element een uniek ID geven
- B. Je kunt stijlen herhalen voor meerdere elementen
- C. Het vervangt de style-tag
- D. Alleen afbeeldingen mogen een class hebben

10. Wat doet de volgende CSS-code?

```
.belangrijk {  
  color: red;  
  font-weight: bold;  
}
```

- A. Alles op de pagina wordt rood en vet
- B. Alleen paragrafen worden rood
- C. Alleen elementen met class="belangrijk" worden rood en vet
- D. Alleen de titel wordt aangepast

11. Welke combinatie is correct voor het opmaken van een afbeelding met een class?

- A. + .plaatje { border: 2px solid black; }
- B. + img { border: plaatje; }
- C. + border: plaatje;
- D. <div plaatje> + img.border { 2px solid black; }

12. Waarom gebruik je margin in CSS?

- A. Voor ruimte *binnen* het element
- B. Om de kleur van de tekst te bepalen
- C. Om ruimte tussen elementen of tot de rand van het scherm te creëren
- D. Om de breedte van een rand te bepalen

13. Welke HTML-structuur is correct voor een basispagina met header, nav, main en footer?

A.

```
<body>  
  <nav></nav>  
  <footer></footer>  
  <main></main>  
  <header></header>  
</body>
```

B.

```
<body>  
  <header></header>
```

```

    <main>
      <nav></nav>
      <footer></footer>
    </main>
  </body>

```

C.

```

<body>
  <header></header>
  <nav></nav>
  <main></main>
  <footer></footer>
</body>

```

D.

```

<html>
  <main></main>
  <header></header>
  <footer></footer>
</html>

```

14. Je wilt drie paragrafen dezelfde opmaak geven. Wat doe je het beste?

- A. Je gebruikt drie keer inline CSS
- B. Je maakt een aparte id per paragraaf
- C. Je maakt één class en gebruikt die bij alle paragrafen
- D. Je zet alle tekst in één <div>

15. Wat doet deze combinatie van HTML en CSS?

HTML:

```

<p class="info">Welkom!</p>

```

CSS:

```

.info {
  font-size: 24px;
  background-color: yellow;
}

```

- A. De tekst wordt klein en geel
- B. De paragraaf wordt groter en krijgt een gele achtergrond
- C. Alle paragrafen krijgen een gele kleur
- D. Alleen <h1>-elementen worden aangepast

LES 5 FLEXBOX

Flexbox is een techniek in CSS waarmee je **elementen netjes kunt uitlijnen en verdelen** binnen een container.

Bijvoorbeeld:

- Elementen **naast elkaar zetten**
- Elementen **centreren** (horizontaal en/of verticaal)
- Ruimte automatisch **verdelen** tussen elementen

Flexbox is gemaakt om **lay-outs eenvoudiger** te maken dan met oudere technieken zoals float of position.

Hoe werkt het?

Flexbox werkt altijd met **twee lagen**:

1. Een **container** → die krijgt display: flex;
2. Eén of meer **child-elementen** → die reageren op die flex-opmaak

Voorbeeld:

HTML

```
<div class="container">
  <div class="vak">1</div>
  <div class="vak">2</div>
  <div class="vak">3</div>
</div>
```

CSS

```
.container {
  display: flex;
}
```

Dit zet de vakjes **naast elkaar**.

Belangrijke Flexbox-eigenschappen

Voor de container (display: flex;):

Eigenschap	Wat het doet
flex-direction	Richting: row (naast elkaar) of column (onder elkaar)
justify-content	Verdeling in de rij (links, midden, ruimte ertussen)
align-items	Uitlijning in de kolom (boven, midden, onder)

Voor de kind-elementen:

Eigenschap	Wat het doet
flex-grow	Laat een item meevergroeien met de ruimte
flex-shrink	Laat een item kleiner worden als er te weinig ruimte is
align-self	Uitlijning van één specifiek item

Voorbeeld: Centreren met Flexbox

HTML

```
<div class="container">
  <div class="vak">Hallo</div>
</div>
```

CSS

```
.container {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 200px;
  background-color: lightgray;
}

.vak {
  background-color: lightblue;
  padding: 20px;
  border: 2px solid navy;
}
```

Wat je ziet: de blauwe "vak" staat **precies in het midden** van het grijze vlak.

OPDRACHT 20. DRIE BLOKKEN NAAST ELKAAR

Je oefent het gebruik van `display: flex` en het uitlijnen van meerdere elementen naast elkaar.

Opdracht

Maak een container met drie vakken erin:

1. De container gebruikt `display: flex`;
2. De vakken staan netjes **naast elkaar**
3. Er zit **ruimte tussen** de vakken
4. Alle vakken zijn even groot en hebben een achtergrondkleur

Extra uitdaging

- Verander de volgorde met `flex-direction: column`;
- Laat één vak groter zijn met `flex: 2`;
- Centreer de hele container op de pagina

OPDRACHT 21. FLEXBOX FROGGY

Speel de Flexbox Froggy game op <https://flexboxfroggy.com/#nl>

OPDRACHT 22. PAGINA-INDELING MET FLEXBOX

Gewenste layout:

```
+-----+
|           header           |
+-----+
|           nav              |
+-----+
| left-blok | main           |
+-----+
|           footer           |
+-----+
```

- header → bovenaan
- nav → eronder, volledige breedte
- dan een rij met left-blok en main naast elkaar
- footer → onderaan

Opdracht

Je gebruikt **HTML + Flexbox** om een pagina te bouwen met duidelijke zones: header, nav, footer én een hoofdgedeelte met een zijbalk.

Eisen

1. header, nav, footer: **volledige breedte**
2. In het hoofdgedeelte (section): gebruik Flexbox om:
 - a. een **left-blok** links te zetten (bijv. 200px breed)
 - b. de **main-content** rechts daarvan te tonen
3. Geef elk blok een eigen kleur en tekst

Extra uitdaging

- Maak de main-section ook responsive op mobiel (verticale stapeling)
- Voeg een afbeelding toe in main en een lijstje in left
- Gebruik gap voor ruimte tussen left en main

LES 6 RESPONSIVE DESIGN

Stel je dit voor:

Je hebt een mooie website gemaakt. Op jouw laptopscherm ziet hij er perfect uit. Maar dan opent iemand dezelfde site op zijn telefoon... en ineens ziet alles er rommelig uit: tekst te groot, blokken te breed, knoppen buiten beeld.

Dat wil je voorkomen.

Daarom gebruiken we **responsive design**: een manier om websites zo te maken dat ze **goed werken op elk schermformaat**, van smartphone tot groot computerscherm.

Wat betekent "responsive"?

Responsive betekent: **meebewegend** of **aanpassend**.

Een responsive website **past zich automatisch aan aan de breedte van het scherm**.

Dat betekent bijvoorbeeld:

- Op een groot scherm staan blokken **naast elkaar**
- Op een klein scherm schuiven die blokken **onder elkaar**
- Tekst wordt **kleiner of groter** afhankelijk van het scherm
- Navigatie verandert van **menubalk** naar **hamburgermenu** (later)

Media queries: de sleutel tot responsive CSS

Een **media query** is een regel in CSS waarmee je zegt:

"Als het scherm een bepaalde breedte heeft, gebruik dan deze speciale opmaak."

Basisvoorbeeld:

```
@media (max-width: 768px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

Dit betekent:

Als het scherm 768 pixels breed is of smaller, geef de achtergrond een lichtblauwe kleur.

Praktischer voorbeeld met Flexbox:

Stel je hebt een left en main blok naast elkaar op groot scherm:

```
.wrapper {  
  display: flex;  
}
```

Dan kun je met een media query zorgen dat ze **onder elkaar komen** op mobiel:

```
@media (max-width: 600px) {  
  .wrapper {  
    flex-direction: column;  
  }  
}
```

}

Wanneer gebruik je een media query?

Gebruik een media query bijvoorbeeld om:

- Kolommen onder elkaar te zetten op mobiel
- Tekst kleiner of groter te maken
- Afbeeldingen op mobiel anders te schalen
- Onderdelen te verbergen of te herschikken

Samenvatting

Term	Uitleg
Responsive design	Layout past zich aan elk schermformaat aan
Media query	CSS-regel die alleen geldt bij bepaalde schermgrootte
Flexibel ontwerp	Blokken schuiven, schalen of veranderen mee



OPDRACHT 23.

MAAK JE WEBSITE RESPONSIVE MET MEDIA QUERIES

Je leert hoe je met een **media query** je website laat meeschalen op kleinere schermen, zoals een telefoon.

Je hebt een webpagina met een header, een main-gedeelte en een left-blok naast elkaar.

Op een **laptop** ziet dat er goed uit.

Maar op een **telefoon** wil je dat de twee blokken **onder elkaar komen te staan**.

Opdracht

1. Voeg **onderaan de CSS** een media query toe:

```
@media (max-width: 700px) {  
  .wrapper {  
    flex-direction: column;  
  }  
  
  .left {  
    width: 100%;  
  }  
}
```

2. Bekijk de pagina in je browser.
3. Verklein het venster (of gebruik Inspecteer > Responsive modus).
4. Controleer of left en main **onder elkaar komen** en even breed worden.

Extra uitdaging

- Voeg nog een media query toe voor tablets (bijv. max-width: 1000px)
- Verklein het lettertype van de header op mobiel

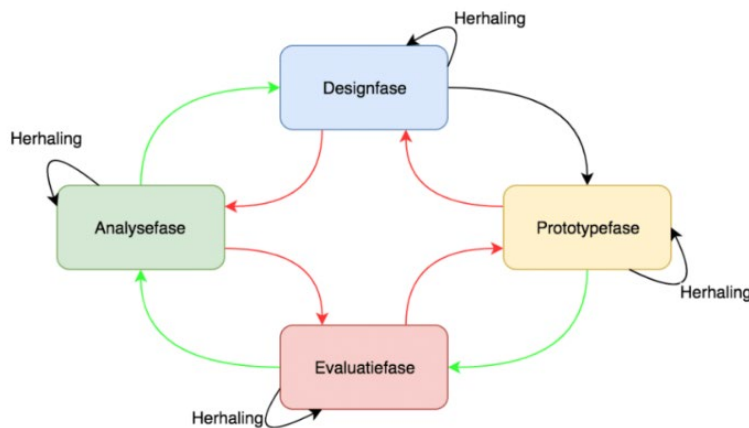
- Voeg een achtergrondkleur toe voor mobiel met:

```
body {  
  background-color: lightgray;  
}
```

LES 7 ONTWERP CYCLUS

Inmiddels heb je geleerd wat usability is en hoe je een goede usability bereikt. Bij het eerste ontwerp van een product moet al rekening gehouden worden met diens usability. Om er zeker van te zijn dat een product een goede usability bereikt, wordt er in het ontwerp process nog minimaal één keer gekeken in hoeverre het product voldoet.

Bij het ontwikkelen van een nieuw product doorloopt een ontwikkelgroep een aantal opeenvolgende stappen, een zogenoemde ontwerp cyclus. Deze cyclus bestaat uit vier fases, die je meerdere keren kan doorlopen. Elke fase heeft andere taken die de ontwikkelgroep uitvoert. De ontwikkeling van een product begint met de analyse fase en eindigt met de evaluatiefase. Na het doorlopen van de cyclus kan de groep na de evaluatie fase beslissen of zij de cyclus nogmaals willen doorlopen, of dat ze klaar zijn. De evaluatiefase is dus bedoeld om te kijken of het product inmiddels goed is.



Figuur 5: De ontwerp cyclus [bron: Lesmodule User Experience]

In de beginfase, de analyse fase, wordt gekeken naar de verschillende stakeholders van het product of systeem. Stakeholders zijn personen of groepen die invloed hebben op het product of systeem, of erdoor worden beïnvloed. Voorbeelden van stakeholders zijn: eindgebruikers, ontwikkelaars, programmeurs en aandeelhouders.

Wanneer de analyse fase is afgerond begint de design fase. In de designfase worden eerst verschillende ideeën ontwikkeld voor het ontwerp van het product of systeem. Na het kiezen van een idee wordt een eerste ontwerp gemaakt. Dit ontwerp kan al getest worden met eventuele eindgebruikers. Wanneer de resultaten van deze test niet goed genoeg zijn kan de ontwikkelgroep ervoor kiezen om de analyse fase nogmaals uit te voeren. Dit is de herhalingspijl bij analysefase in de figuur hierboven.

Na het ontwikkelen van een goed ontwerp volgt de prototype fase. In deze fase wordt een prototype van het product of systeem gemaakt. Dit kan heel laagdrempelig zijn, zoals een schets op papier, maar ook wat meer geavanceerd zoals een prototype op de computer of een 3D-print.

In de evaluatie fase wordt het prototype getest door eindgebruikers. Het evalueren van een product is belangrijk omdat je problemen kan ontdekken en verhelpen. Daarnaast kunnen bij de evaluatie

nieuwe inzichten naar boven komen en kan er over de gebruikers geleerd worden. De evaluatiefase moet altijd doorlopen worden, ook door ervaren ontwikkelaars. Dit is omdat niemand precies kan voorspellen hoe gebruikers gaan reageren. De behoefte van de gebruiker kan zelfs tegen de intuïtie van de ontwerper ingaan.



Ontdek problemen



Ontdek kansen



Leer gebruikers kennen

Figuur 6: De voordelen van het evalueren van een product.



OPDRACHT 24. TERUGBLIK EERSTE LESSEN

Beredeneer met welke fase we in de afgelopen weken bezig zijn geweest.



OPDRACHT 25. METHODES ZOEKEN

Deze les heb je onder andere geleerd over de evaluatiefase. Nu ga je nadenken over de taken die een ontwikkelgroep in deze evaluatiefase uitvoert.

Zoek op het internet twee methodes op om te testen of de usability voor een product goed is. Gebruik hiervoor de zoekterm '*usability testen*' of '*usability testing*'. Een voorbeeld van een methode om usability te testen is eyetracking.

Vul voor beide methodes de volgende vragenlijst in:

Voor welk product gebruik je deze test?

Wie worden getest?

Wie nemen de test af?

Wat wordt getest?

Hoe wordt getest?

Wat is een voorbeeld van een resultaat uit de test?

Waarom is het resultaat nuttig om te weten?

Voor welk product gebruik je deze test?

Wie worden getest?

Wie nemen de test af?

Wat wordt getest?

Hoe wordt getest?

Wat is een voorbeeld van een resultaat uit de test?

Waarom is het resultaat nuttig om te weten?



OPDRACHT 26. METHODES

Deze opdracht maak je weer op maken.mijn-in.nl.

Deze les heb je onder andere geleerd over de evaluatiefase. Nu ga je nadenken over de taken die een ontwikkelgroep in deze evaluatiefase uitvoert.

Bedenk zelf twee methodes om te testen of de usability voor jouw te onderzoeken website goed is. (zie de eerdere antwoorden op maken.mijn-in.nl).

Vul voor beide methodes de volgende vragenlijst in:

Voor welk product test je? (zie opdracht 3)

Wie worden getest?

Wie nemen de test af?

Wat wordt getest?

Hoe wordt getest?

Wat is een voorbeeld van een resultaat uit de test?

Waarom is het resultaat nuttig om te weten?

Let op de antwoorden die je hier kan geven zijn wel erg beperkt. Dit moet je op maken.mijn-in.nl verder uitwerken. Je mag ook gebruik maken van de lessen 10 t/m 10d bij het beantwoorden van de vragen.

LES 8 FUNCTIONEEL ONTWERP

Inleiding op het functioneel ontwerp van een website

Voordat een website gemaakt wordt met HTML en CSS, is het belangrijk om eerst goed na te denken over wat de website moet doen en hoe deze eruit moet zien. Dit proces begint met een *functioneel ontwerp*. Een functioneel ontwerp is een beschrijving van wat een website moet kunnen en hoe de gebruiker ermee omgaat. Je kunt het zien als een blauwdruk of plan voordat je begint te bouwen. Net zoals een architect eerst een bouwtekening maakt voordat een huis wordt gebouwd, maak je bij een website eerst een functioneel ontwerp voordat je gaat programmeren.

Een functioneel ontwerp beschrijft niet *hoe* iets technisch gemaakt wordt, maar *wat* er moet gebeuren. Het richt zich op de functies van de website: wat kunnen bezoekers doen? Welke informatie moeten ze kunnen vinden? Hoe navigeer je van de ene pagina naar de andere? Denk bijvoorbeeld aan een webshop: bezoekers moeten producten kunnen zoeken, informatie kunnen lezen, producten in een winkelmandje kunnen doen en afrekenen. Al deze dingen worden opgenomen in het functioneel ontwerp.

Daarnaast is het belangrijk om na te denken over de structuur van de website. Welke pagina's zijn er? Hoe hangen die met elkaar samen? Dit noem je de *navigatiestructuur* of de *site-structuur*. Vaak wordt dit visueel gemaakt met een *sitemap*, een soort boomdiagram waarin je ziet hoe de pagina's zijn opgebouwd. Een duidelijke structuur helpt gebruikers om snel te vinden wat ze zoeken.

Het functioneel ontwerp bevat ook vaak een *wireframe* of *schets* van hoe de pagina's eruit moeten zien. Dit is nog geen mooi ontwerp met kleuren en afbeeldingen, maar een simpele weergave van de indeling van de pagina. Denk aan blokken waar de titel, het menu, de tekst, afbeeldingen en knoppen komen te staan. Deze wireframes helpen om te bedenken hoe je de inhoud logisch kunt indelen. In deze fase beslis je bijvoorbeeld of het menu bovenaan of aan de zijkant komt, en of er een zoekbalk aanwezig is.

Voor leerlingen die nog nooit met HTML en CSS hebben gewerkt, is het functioneel ontwerp een handige eerste stap. Je hoeft nog niet te weten hoe je iets moet coderen, maar je denkt alvast na over *wat* je gaat maken en *waar* alles komt te staan. Dat maakt het later makkelijker om een website te bouwen, omdat je een duidelijk plan hebt.

In deze lessenreeks ga je leren hoe je van een idee naar een eenvoudige website gaat. Je begint met het bedenken van wat je site moet doen (functioneel ontwerp), daarna leer je hoe je de structuur opbouwt met HTML (de inhoud en opbouw van de pagina), en vervolgens hoe je met CSS de vormgeving en stijl bepaalt (zoals kleuren, lettertypes en indeling). Door eerst goed na te denken over het ontwerp, maak je websites die overzichtelijk, duidelijk en prettig te gebruiken zijn voor iedereen. Dat is het doel van een goed functioneel ontwerp.

OPDRACHT 27. FUNCTIONEEL ONTWERP START

Jullie gaan samen in tweetallen werken aan het functioneel ontwerp van een website. Dat betekent dat je eerst goed nadenkt over wat jullie website moet kunnen en voor wie je deze maakt, voordat je begint met bouwen in HTML en CSS.

Werkwijze

- Werk samen in een gedeeld **Google Docs-document**.
- Zet **op de eerste pagina jullie voor- en achternaam** duidelijk bovenaan.
- Deel het document met de docent (de juiste e-mail krijg je van je docent).
- Gebruik de uitleg en opdrachten hieronder om alle hoofdstukken uit te werken.

Zorg dat jullie antwoorden duidelijk en in volledige zinnen zijn geschreven. De tekst moet begrijpelijk zijn voor iemand die jullie website nog niet kent.

Je maakt een reken, taal of topografie website waar kinderen van de basisschool kunnen oefenen. De website hoeft niet te werken. In de eerste praktische opdracht in les 1 heb je al een aantal bestaande websites bekeken.

Veel succes!

HOOFDSTUK 1 DOELEN EN DOELGROEP

Voordat je een website gaat ontwerpen, moet je eerst goed nadenken *waarvoor* je de website maakt (het doel) en *voor wie* je dat doet (de doelgroep). Dit is de basis van elk functioneel ontwerp. Als je dit goed hebt uitgewerkt, kun je veel beter beslissen hoe de website eruit moet zien en welke functies hij nodig heeft.

1. Het doel van de website

Het doel beschrijft wat je met de website wilt bereiken. Dit kan informatief zijn (bijvoorbeeld een website over gezonde voeding), maar ook commercieel (producten verkopen) of praktisch (een inschrijfformulier voor een sportclub). Denk goed na: *waarom bestaat deze website eigenlijk? Wat moet een bezoeker hier kunnen doen of leren?*

Voorbeelden van doelen:

- Bezoekers informeren over een bepaald onderwerp.
- Mensen laten inschrijven voor een activiteit.
- Producten of diensten aanbieden.
- Inspireren of vermaken met foto's, verhalen of video's.

2. De doelgroep

De doelgroep is de groep mensen voor wie je de website maakt. Denk aan leeftijd, interesses, taalgebruik, digitale ervaring en wat ze verwachten van de site. Een website voor kinderen ziet er heel anders uit dan een site voor ouderen of voor een sportvereniging.

Voorbeelden van doelgroepen:

- Kinderen van 8 tot 12 jaar die informatie zoeken voor een spreekbeurt.
- Tieners die zich willen inschrijven voor een schooltoernooi.
- Ouders die informatie zoeken over gezonde voeding.
- Fans van een band die concertdata willen zien.

Als je weet voor wie je de website maakt, kun je betere keuzes maken over taalgebruik, kleuren, lettergrootte, navigatie en inhoud.



OPDRACHT 28. DOEL EN DOELGROEP

Schrijf een korte beschrijving van het doel en de doelgroep van jouw website. Gebruik de volgende vragen als hulp:

A. Het doel van je website

1. Wat wil je met de website bereiken?
2. Wat moet een bezoeker kunnen doen op de site?
3. Waarom is deze site nuttig of belangrijk?

Schrijf dit in 3 tot 5 zinnen.

B. De doelgroep van je website

1. Voor wie is de website bedoeld? (leeftijd, interesses, bijvoorbeeld)
2. Wat weet deze doelgroep al over het onderwerp?
3. Waarmee moet je rekening houden bij het ontwerpen voor deze doelgroep?

Schrijf dit ook in 3 tot 5 zinnen.

Voorbeeld:

Doel:

Deze website is bedoeld om jongeren tussen de 12 en 16 jaar te informeren over mentale gezondheid. Bezoekers kunnen tips lezen, video's bekijken en testjes doen om meer te leren over hoe ze zich voelen. Het doel is om jongeren te helpen zich beter te voelen en te laten weten waar ze hulp kunnen vinden.

Doelgroep:

De doelgroep is middelbare scholieren van 12 tot 16 jaar. Ze zijn gewend om veel op hun telefoon te zitten en korte, duidelijke informatie te lezen. De site moet dus mobielvriendelijk zijn, vlot geschreven teksten hebben en er aantrekkelijk uitzien met beeld en kleur.

HOOFDSTUK 2 LIJST MET EISEN EN WENSEN

In dit hoofdstuk ga je bepalen **wat je website minimaal moet kunnen (de eisen)** en **wat je daarnaast graag zou willen (de wensen)**. Dit helpt je om overzicht te houden tijdens het bouwen van je website en om prioriteiten te stellen: eerst zorgen dat de basis werkt, daarna kun je het mooier of uitgebreider maken.

Wat zijn eisen?

Eisen zijn functies of onderdelen die *verplicht* zijn voor jullie website. Zonder deze dingen werkt de site niet goed of mist hij zijn doel. Eisen zijn dus de dingen die echt móéten worden gemaakt.

Voorbeelden van eisen:

- De website heeft een homepage en minstens twee andere pagina's.
- Er is een duidelijk menu om tussen pagina's te navigeren.
- De site werkt op mobiel en laptop.
- Er staat informatie op over het onderwerp van de site.
- Gebruikers kunnen contact opnemen via een formulier.

Wat zijn wensen?

Wensen zijn dingen die *extra* zijn. Ze maken de website mooier, leuker of handiger, maar zijn niet perse nodig om de website bruikbaar te maken. Als je tijd hebt of extra uitdaging zoekt, kun je deze dingen later toevoegen.

Voorbeelden van wensen:

- Er zit een bewegende banner bovenaan de pagina.
- De site heeft een donkere modus (dark mode).
- Er worden video's of animaties getoond.
- Er zit een interactief onderdeel in, zoals een quiz.
- De teksten zijn beschikbaar in meerdere talen.

Door een lijst te maken met eisen én wensen, kun je je project goed plannen en weet je wat belangrijk is om eerst te maken.

OPDRACHT 29. EISEN EN WENSEN

Maak twee lijsten in jullie document:

1. Lijst met eisen (minimaal 5 punten):

- Wat moet je website minimaal hebben om goed te functioneren?
- Schrijf elke eis als een korte, duidelijke zin.
- Begin met de belangrijkste onderdelen die je al zeker weet.

2. Lijst met wensen (minimaal 3 punten):

- Wat zou je graag toevoegen als er tijd en ruimte is?
- Kies dingen die je interessant vindt, maar die niet verplicht zijn.
- Laat ook zien wat je creatief of uitdagend vindt.

Voorbeeld:

Lijst met eisen:

- De website heeft een homepage, een 'over ons'-pagina en een contactpagina.
- Het menu staat op elke pagina en werkt goed.
- De site werkt op mobiel en computer.
- De hoofdkleur past bij het onderwerp van de site.
- De gebruiker kan makkelijk terug naar de homepage.

Lijst met wensen:

- Er is een video op de homepage.
- De achtergrond verandert mee met het tijdstip van de dag.
- Er zit een quiz in over het onderwerp van de site.

Let op: Schrijf de lijst samen in hetzelfde Google Docs-document, **onder hoofdstuk 2**. Zorg dat de lijsten overzichtelijk zijn en goed leesbaar voor iemand die nog niks weet van jullie site.

HOOFDSTUK 3 USER STORIES

Een **user story** is een korte, simpele manier om te beschrijven wat een bezoeker op jullie website wil doen. Je bekijkt je website dus niet vanuit jezelf als maker, maar vanuit de **gebruiker**. Wat wil die persoon op jullie website doen of bereiken?

Een goede user story helpt je om na te denken over de functies die belangrijk zijn en hoe de website logisch werkt voor de gebruiker. Dat zorgt ervoor dat je website duidelijk en bruikbaar is.

Een user story heeft altijd dezelfde opbouw:

Als [type gebruiker] wil ik [wat diegene wil doen], zodat [waarom].

Hiermee geef je aan:

- **Wie** de gebruiker is.
- **Wat** diegene wil doen op de website.
- **Waarom** dat belangrijk is of wat het oplevert.

Voorbeeld user stories:

- Als **nieuwe bezoeker** wil ik **snel zien waar de website over gaat**, zodat **ik weet of het interessant voor mij is**.
- Als **ouder** wil ik **informatie vinden over de lestijden**, zodat **ik mijn kind op tijd kan ophalen**.
- Als **fan** wil ik **foto's van het laatste optreden kunnen bekijken**, zodat **ik herinneringen kan ophalen**.
- Als **leerling** wil ik **mij inschrijven voor het schooltoernooi**, zodat **ik mee kan doen aan de wedstrijden**.

Goede user stories zijn kort, duidelijk en gericht op wat de gebruiker wil, niet op de techniek. Dat helpt om keuzes te maken tijdens het ontwerpen.

OPDRACHT 30. USER STORIES

Schrijf minimaal 4 user stories voor jullie website.

- Gebruik steeds de structuur:
"Als [type gebruiker] wil ik [actie], zodat [doel]."
- Denk aan verschillende soorten gebruikers: jong/oud, bekend/nieuw, actief/passief.
- Zorg dat je user stories echt passen bij het doel en de doelgroep van jullie site.

Zet de user stories netjes onder elkaar in jullie document, onder **hoofdstuk 3**.

Voorbeeld (voor een site over een schooltoernooi):

User stories:

- Als deelnemer wil ik het wedstrijdschema kunnen zien, zodat ik weet wanneer ik moet spelen.
- Als ouder wil ik foto's van het toernooi kunnen bekijken, zodat ik mijn kind kan terugzien.
- Als docent wil ik zien hoeveel leerlingen zich al hebben aangemeld, zodat ik weet of er genoeg teams zijn.
- Als bezoeker wil ik informatie over de locatie kunnen vinden, zodat ik weet waar het toernooi gehouden wordt.

Let op: Gebruik jullie eerdere hoofdstukken als basis. De user stories moeten logisch passen bij wat je in hoofdstuk 1 en 2 hebt beschreven. Zo bouw je stap voor stap aan een goed functioneel ontwerp.

HOOFDSTUK 4 STRUCTUUR

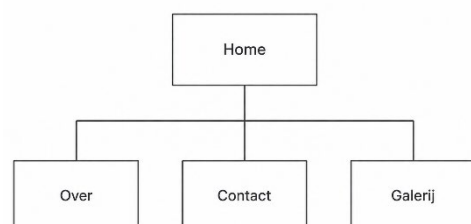
Nu jullie weten *wat* de website moet doen en *voor wie*, is het tijd om na te denken over de **structuur** van de website. Daarmee bedoelen we:

- Hoe zit de website in elkaar?
- Uit welke pagina's bestaat de site?
- Hoe navigeert een gebruiker van de ene naar de andere pagina?

De structuur is als het geraamte van de website: de basisvorm waarop je verder bouwt. Dit is een belangrijke stap in je functioneel ontwerp. Een goede structuur maakt de website logisch en makkelijk te gebruiken.

Een website heeft bijna altijd meerdere pagina's. De homepage is meestal het startpunt. Van daaruit klik je door naar andere pagina's, zoals "Over ons", "Contact", of "Galerij". Al die pagina's vormen samen de **navigatiestructuur** of **sitestructuur**.

Om deze structuur duidelijk te maken, maak je een **diagram** waarin je laat zien hoe de pagina's met elkaar verbonden zijn. Dat heet een **sitemap**. Een voorbeeld staat hiernaast.



Een sitemap is geen tekening van hoe de site eruitziet, maar een schema van de opbouw. Later (in hoofdstuk 6) ga je pas ontwerpen hoe de pagina's er visueel uitzien. Hier gaat het alleen om de *structuur* en de *relatie tussen pagina's*.

Belangrijk:

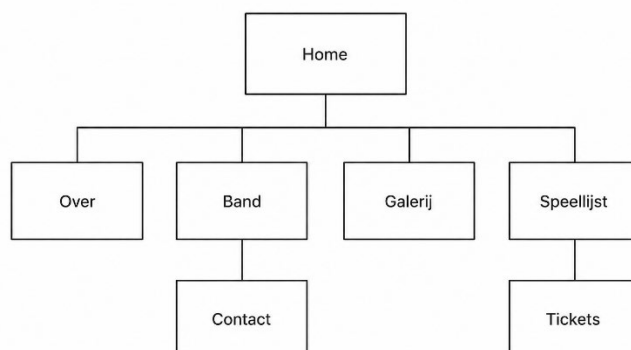
- De structuur moet **digitaal** worden uitgewerkt. Dus géén foto's van tekeningen op papier.
- Je mag dit maken in Google Drawings, Lucidchart, Canva, een tekentool, of gewoon met vormen en pijlen in Google Docs of Slides.

OPDRACHT 31. SITEMAP

Maak een duidelijke structuur (sitemap) van jullie website.

- Begin met de **homepage**.
- Teken daarna alle andere pagina's die bij jullie site horen.
- Geef met pijlen of lijnen aan hoe je van de ene pagina naar de andere gaat.
- Zorg dat het overzicht **digitaal** en **netjes leesbaar** is.
- Zet de structuur in jullie Google Docs-document, onder **hoofdstuk 4**. Je mag het schema hierin plakken of als afbeelding invoegen als je het in een andere tool hebt gemaakt.

Voorbeeld structuur voor een muzieksite:



Tip: Kijk nog eens naar je user stories uit hoofdstuk 3. Welke pagina's heb je nodig om die mogelijk te maken? Gebruik die informatie bij het opbouwen van je structuur.

Zo zorg je ervoor dat je website straks logisch werkt en goed aansluit op wat gebruikers willen doen.

HOOFDSTUK 5 BASISELEMENTEN

In dit hoofdstuk ga je nadenken over de **vaste onderdelen** en de **uitstraling** van je website. Je beschrijft welke onderdelen op elke pagina terugkomen en hoe je website er ongeveer uit moet gaan zien qua sfeer, kleuren en stijl. Dit helpt je straks bij het maken van een duidelijke en aantrekkelijke website.

Basiselementen

Je website bestaat uit onderdelen die op (bijna) elke pagina terugkomen. Denk aan:

- **Header:** het bovenste deel van de site, meestal met de titel of het logo.
- **Navigatie(menu):** hiermee klik je naar andere pagina's (zoals 'Home' of 'Contact').
- **Main:** het hoofdgedeelte van de pagina, waar de inhoud komt te staan.
- **Footer:** onderaan de pagina, met bijvoorbeeld contactgegevens of links naar social media.
- **Eventueel een zijbalk:** met extra info of knoppen (optioneel).

Door vaste elementen steeds op dezelfde plek te gebruiken, wordt de site overzichtelijk en prettig in gebruik.

Sfeer en uitstraling

Naast de structuur is ook de sfeer belangrijk. Je denkt na over:

- **Kleurgebruik:** Welke kleuren passen bij het onderwerp en de doelgroep? Kies 2 tot 4 hoofdkleuren die je overal terug laat komen.
- **Lettertypes:** Moet het strak en modern zijn, of juist speels en creatief?
- **Sfeerwoorden:** Bijvoorbeeld: vrolijk, stoer, rustig, zakelijk, sportief. Je kan hier ook gebruik maken van plaatjes die de sfeer weergeven.

Een heldere stijl helpt om een goede indruk te maken en past bij het doel van je site.

Logo

Heeft je website een logo of beeldmerk? Dan kun je ook alvast nadenken over het ontwerp daarvan:

- Wat stelt het logo voor?
- Welke kleuren en vormen gebruik je?
- Waar op de pagina komt het logo te staan?

Je hoeft het logo nog niet in detail te ontwerpen, maar geef wel aan of je er één wilt gebruiken en waar. Je kan dit natuurlijk ook visueel laten zien.

OPDRACHT 32. BASISELEMENTEN

Maak een overzicht van de basiselementen en uitstraling van jullie website. Zet dit in jullie document onder hoofdstuk 5. Let op in dit hoofdstuk is 'laten zien' (illustreeren) ook een belangrijk onderdeel van de beschrijving.

1. Beschrijf de vaste onderdelen die op elke pagina voorkomen:

- Header, navigatie, main, footer, eventueel een zijbalk.
- Geef per onderdeel kort aan wat erin komt te staan.

2. Beschrijf de sfeer en uitstraling van jullie site:

- Kies 2–4 hoofdkleuren.
- Geef aan welk lettertype je ongeveer wilt gebruiken (bijvoorbeeld: strak, speels, klassiek).
- Schrijf 3 sfeerwoorden die passen bij jullie site.

3. Beschrijf kort of je een logo wilt gebruiken:

- Wat stelt het logo voor?
- Waar komt het te staan op de pagina?
- (Je mag ook alvast een schetsje maken als je wilt.)

Voorbeeld (samengevat):

- **Header:** bevat het logo links en de titel van de site.
- **Navigatie:** horizontaal onder de header met knoppen naar 4 pagina's.
- **Main:** in het midden, verandert per pagina.
- **Footer:** contactgegevens en Instagram-link.

Sfeer en kleuren:

- Hoofdkleuren: donkerblauw, wit, lichtgrijs.
- Lettertype: strak en modern.
- Sfeerwoorden: professioneel, rustig, duidelijk.

Logo:

- Voorstelling: een gestileerde vogel als symbool voor vrijheid.
- Locatie: linksboven in de header.

Door deze keuzes vast te leggen, leg je een stevige basis voor je ontwerp. Je weet straks niet alleen *wat* je gaat maken, maar ook *hoe het eruit moet zien*. Dat maakt het bouwen in HTML en CSS veel makkelijker!

LOGO (voorbeeld)



Het logo staat in de header links, zodat het altijd zichtbaar is.

KLEURENPALET (2-4 hoofdkleuren)

#1E2A38

#4CAF9A

#F39C3D

#F7F6F2

Deze kleuren geven een natuurlijke, frisse en betrouwbare uitstraling.

SFEER / SFEERWOORDEN



Avontuurlijk • Rustgevend • Natuurlijk • Betrouwbaar

LETTERTYPES (voorbeeld)

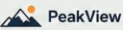
H1 Koptekst

Subkoptekst

Dit is een voorbeeld van lopende tekst.
Dit lettertype is goed leesbaar en past bij de uitstraling van de site.

Aanbevolen lettertypes:
Poppins (kopteksten)
Open Sans (lopende tekst)

PAGINA-INDELING (voorbeeld wireframe)

Home Over Bestemmingen Galerij Contact



Welkom bij PeakView

Ontdek de mooiste plekken in de natuur en beleef jouw volgende avontuur.

[Bekijk bestemmingen](#)

Over PeakView

Wij inspireren avonturiers om de natuur te ontdekken.

Snelle links

- Home
- Over
- Bestemmingen
- Contact

Volg ons

Zijbalk (optioneel)
Extra informatie, nieuws of knoppen.

HEADER
Bevat logo en navigatiemenu.

MAIN
Hoofdinhoud van de pagina.

FOOTER
Extra informatie en links naar social media.

ZIJBALK (OPTIONEEL)
Voor aanvullende inhoud.

- 67 -

HOOFDSTUK 6 SCHERMONTWERP

In dit laatste hoofdstuk van het functioneel ontwerp ga je een **schermontwerp** maken. Dat is een visuele uitwerking van hoe jullie website eruit gaat zien. Je laat hierin zien waar teksten, knoppen, afbeeldingen en andere onderdelen precies komen te staan op de verschillende pagina's.

Je hebt in hoofdstuk 4 al de **structuur** bepaald (welke pagina's er zijn), en in hoofdstuk 5 de **basiselementen en stijl**. Nu ga je dat verder uitwerken in een ontwerp per pagina. Dit wordt ook wel een **wireframe** genoemd: een eenvoudige schets van de indeling van je webpagina's, zonder dat het al helemaal opgemaakt of "mooi" is.

Een schermontwerp helpt om:

- Te bedenken *wat* er op elke pagina komt te staan.
- Te bepalen *waar* elk onderdeel komt (bijvoorbeeld: komt het menu boven of naast de inhoud?).
- Te zorgen voor een logische en aantrekkelijke indeling.

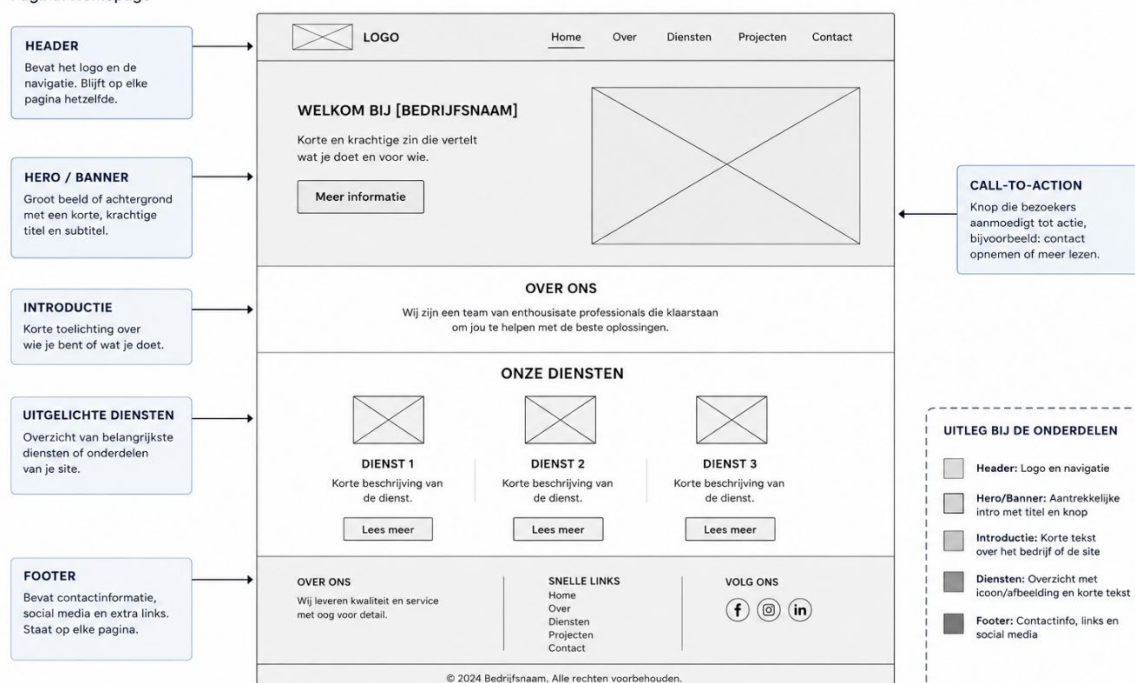
Je kunt het zien als het plan voordat je met HTML en CSS echt gaat bouwen.

Wat zet je in een schermontwerp?

- Alle vaste onderdelen zoals header, navigatie, main en footer.
- De plaats van tekst, afbeeldingen, knoppen, video's of formulieren.
- De indeling van elk type pagina (bijvoorbeeld: de homepage is anders dan de contactpagina).
- Een *simpele* aanduiding van inhoud (bijvoorbeeld: een rechthoek met "Afbeelding" of "Titel").

VOORBEELD: SCHERMONTWERP (WIREFRAME)

Pagina: Homepage



Je hoeft het schermontwerp niet kleurrijk of grafisch mooi te maken. Het gaat om de **opbouw en inhoud**, niet om het grafisch ontwerp (dat komt later met CSS).

OPDRACHT 33. SCHERMONTWERP

Maak voor één pagina een schermontwerp.

1. Kies de belangrijke pagina van je website (bijvoorbeeld: de homepage).
2. Maak een **wireframe**: een eenvoudig, digitaal ontwerp van de indeling.
 - Gebruik blokken en tekstlabels om aan te geven wat waar komt (bijv. “Titel”, “Afbeelding”, “Knop naar contactformulier”).
 - Geef vaste onderdelen zoals header, menu en footer duidelijk weer.
 - Houd je aan de sfeer en structuur die je in de vorige hoofdstukken hebt bepaald.
3. Werk het ontwerp **digitaal uit** (niet op papier!). Je mag hiervoor:
 - Vormen en tekstvakken gebruiken in Google Docs of Slides,
 - Of een andere digitale tool gebruiken, zoals Google Drawings, Canva of Figma.

Zet beide schermontwerpen onder hoofdstuk 6 in jullie Google Docs-document.

Je mag de ontwerpen rechtstreeks in het document maken of als afbeelding invoegen.

OPDRACHT 34. AFRONDING VAN HET FUNCTIONEEL ONTWERP

Jullie hebben nu stap voor stap gewerkt aan een compleet functioneel ontwerp voor een website. In dit document heb je nagedacht over het doel van je site, de gebruiker, de inhoud, de structuur én het uiterlijk. Het is nu tijd om alles netjes af te ronden en klaar te maken voor inlevering.

Volg de checklist hieronder om je document compleet, netjes en overzichtelijk te maken.

1. Voorkant (titelpagina)

Maak een nette eerste pagina met de volgende informatie:

- **Titel:** Functioneel ontwerp van [naam van jullie website]
- **Namen van de makers:** Voor- en achternaam van beide leerlingen
- **Klas**
- **Vak / docent**
- **Datum van inleveren**

Je mag de voorkant rustig wat opmaken, bijvoorbeeld met een passend lettertype, een lijn, icoon of rustige achtergrondkleur — maar houd het overzichtelijk en professioneel.

2. Inhoudsopgave

Voeg op de tweede pagina een **inhoudsopgave** toe.

- Gebruik de titels van de zes hoofdstukken:
 1. Doelen en doelgroep
 2. Lijst met eisen en wensen
 3. User stories
 4. Structuur
 5. Basiselementen
 6. Schermontwerp
- Zorg dat de paginanummers erbij staan.
- In Google Docs kun je automatisch een inhoudsopgave invoegen via **Invoegen > Inhoudsopgave**.

3. Paginanummers

- Voeg paginanummers toe onderaan de pagina's.
- In Google Docs: **Invoegen > Paginanummers > Kies onderaan (footer)** en begin pas bij de inhoud (niet op de titelpagina).

4. Indeling en opmaak controleren

- Kijk of elk hoofdstuk duidelijk een **kopje met het hoofdstuknummer en de titel** heeft.
- Zorg dat de tekst goed leesbaar is (gebruik bijvoorbeeld lettertype Arial of Calibri, grootte 11 of 12).
- Controleer of er geen losse zinnen of halve stukken ontbreken.
- Let op **spelling en zinsbouw**: het moet er verzorgd uitzien.
- Zorg dat alle afbeeldingen, diagrammen of schermontwerpen netjes zijn ingevoegd en goed leesbaar zijn.

Door deze afronding serieus te nemen, lever je een functioneel ontwerp in dat niet alleen inhoudelijk goed is, maar er ook professioneel uitziet. Dat helpt je straks bij het bouwen van een goede website én laat zien dat je zorgvuldig werkt.

LES 9 PROTOTYPE MAKEN

OPDRACHT 35. WEBSITE VOLGENS FUNCTIONEEL ONTWERP

Je maakt een werkende website met HTML en CSS, gebaseerd op je eigen functioneel ontwerp (FO). Je gebruikt hiervoor de kennis uit de lessen over structuur, Flexbox, opmaak en semantische HTML. Je werkt deze uit in maken.mijn-in.nl op de server.

Stap 1: Functioneel ontwerp doornemen

- Lees je eigen FO door.
- Controleer of je in elk hoofdstuk duidelijk hebt beschreven wat je nodig hebt om je site te bouwen:
 - Heb je een logische structuur (hoofdstuk 4)?
 - Weet je welke blokken op elke pagina moeten staan (hoofdstuk 5)?
 - Is het schermontwerp concreet (hoofdstuk 6)?

Stap 2: Bouw de HTML-structuur van je pagina's

Doel: Zet de inhoud op de pagina's zonder opmaak.

Opdracht:

- Maak één hoofdpagina (index.html).
- Gebruik semantische HTML: <header>, <nav>, <main>, <footer>, <section>, <article>...
- Voeg vaste onderdelen toe zoals navigatie, titels, tekst, blokken voor afbeeldingen, enz.
- Voeg extra pagina's toe als dat in je structuur staat (bijv. contact.html, over.html).
- Zorg dat je links tussen de pagina's werken.

Tip: Voeg in elk blok tijdelijk een tekstje toe zoals <!-- hier komt het contactformulier -->.

Stap 3: Voeg CSS toe en gebruik Flexbox voor de layout

Doel: Zorg dat je site er goed uit ziet én netjes is ingedeeld.

Opdracht:

- Koppel een CSS-bestand aan je HTML-pagina's
- Gebruik Flexbox om blokken naast elkaar te zetten (zoals left + main)
- Pas het box model toe: margin, padding, border
- Gebruik kleuren, lettertypes, witruimte en uitlijning om je ontwerp te volgen

Eisen:

- Navigatiebalk die er op elke pagina hetzelfde uit ziet
- Een lay-out waarin main en een zijblok naast elkaar staan
- Passende kleuren, lettertypes en afstanden volgens je FO

Stap 4:

Doel: Zorg dat de website responsive is.

Opdracht:

- Gebruik media queries om je website ook te laten werken op een telefoonscherm.

Eisen:

- Alle blokken passen logisch in het kleine scherm
- Je mag ook blokken laten verdwijnen of op een heel andere plek tonen in je pagina

Stap 5: Test je site op inhoud en werking

Doel: Werkt je site zoals bedoeld?

Opdracht:

- Klik door alle links en check of alles zichtbaar is
- Controleer of je ontwerp klopt met je FO
- Vraag een klasgenoot om je site te bekijken met de vraag: "Snap jij wat deze site wil doen?"

Checklist:

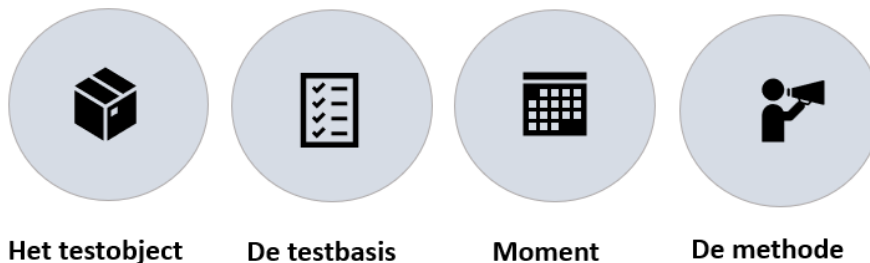
- Sluit alles aan op doelen, doelgroep en eisen?
- Zijn de teksten leesbaar en goed verdeeld?
- Werkt alles ook als je het scherm smaller maakt?

LES 10 EVALUATIE METHODES

In het vorige hoofdstuk hebben we gekeken naar de design cyclus. De heuristieken die eerder behandeld zijn worden in de designfase gebruikt. Vervolgens wordt een prototype gebouwd, daarop gaan we in deze lesmodule niet verder in. We gaan wel dieper in op de evaluatiefase, namelijk het onderzoeken van hoe goed de usability van een product is. Om dit te kunnen bepalen hebben we evaluatiemethoden nodig. Deze evaluatie methoden worden in de volgende hoofdstukken behandeld. We werken één evaluatiemethode uit op je website die je gekozen hebt in opdracht 3. En we werken een evaluatiemethode uit op je eigen ontwikkelde website.

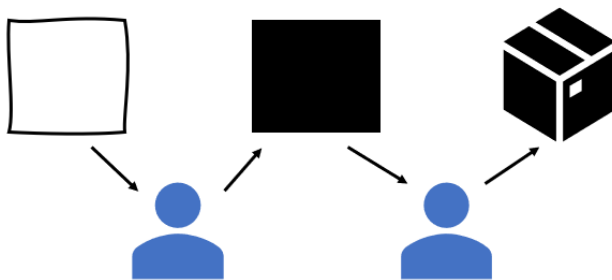
Testen

Testen is bedoeld om vast te stellen in hoeverre het product aan de eisen voldoet. Hierbij is het belangrijk om te weten wat er getest gaat worden (*het testobject*), de eisen (*de testbasis*), wanneer er getest gaat worden (*moment*) en hoe er getest gaat worden (*de methode*).



Figuur 7: Belangrijke elementen van een test.

Vaak wordt *iteratief testen* toegepast. Iteratief testen betekent, dat een product meerdere keren achter elkaar wordt getest en na elke test op basis van de verkregen resultaten verder wordt aangepast. Het product wordt dus iedere keer (een stukje) verbeterd. Hiermee worden usability problemen opgelost voordat het product wordt uitgebracht. Het resultaat hiervan is een product dat niet in één keer goed hoeft te zijn, maar na het eerste ontwerp nog kleine veranderingen ondergaat.



Figuur 8: Een weergave van de ontwikkeling van een doos icoontje als gevolg van iteratief testen.

In figuur 8 zie je een voorbeeld van het iteratief testen van een doos icoontje en de voorgenomen aanpassingen. Hierbij worden er drie versies ontwikkeld. De eerste versie is een wit blokje. Deze wordt getest en op basis van die resultaten is er een tweede versie ontwikkeld. Deze versie is al een stuk beter. Na het testen van de tweede versie is in dit geval het perfecte eindontwerp bereikt.

Subjectieve en objectieve evaluatiemethoden

Subjectief en objectief zijn twee manieren van kijken naar bepaalde dingen. Objectief kijken betekent dat je kijkt naar dingen zoals ze zijn. De resultaten van objectieve evaluatie zijn meetbaar. Bij het uitvoeren van dezelfde objectieve test op hetzelfde product zou iedereen op hetzelfde resultaat moeten komen. Bij objectief meten krijg je alleen objectieve gegevens en dus geen meningen. Dit kan genoeg zijn voor het onderzoek, maar als dit niet het geval is kan de onderzoeker niet doorvragen bij de deelnemers. Subjectief kijken is een evaluatie of reactie op het objectieve kijken. Bij subjectief evalueren gaat het om je mening. Wanneer meerdere onderzoekers een zelfde subjectieve evaluatie uitvoeren op hetzelfde product, kan het zijn dat de resultaten verschillen. Meninge kunnen onvolledig of niet representatief zijn. Subjectief meten brengt wel de mogelijkheid om door te vragen bij de deelnemers. Beide methodes hebben dus voor- en nadelen.



OPDRACHT 36.

HUISWERK TESTOBJECTEN

Lees onderstaande tekst en beantwoord de vragen over testobject, testbasis, moment en methode.

Tim moet voor een klant een software programma ontwerpen. De klant heeft van te voren een aantal eisen aan de usability het design gesteld en dit in een document aangeleverd. Voorbeelden van zulke eisen zijn “Het programma moet niet onnodig complex zijn” en “Het programma moet gemakkelijk te gebruiken zijn” Om er zeker van te zijn dat het programma een goede usability bereikt, voert Tim een SUS test uit na het afmaken van zijn eerste versie voor het ontwerp.

Wat is in bovenstaand verhaal het testobject? _____

En de testbasis? _____

En het moment? _____

En de methode? _____

LES 10A SUBJECTIEVE EVALUATIE METHODEN

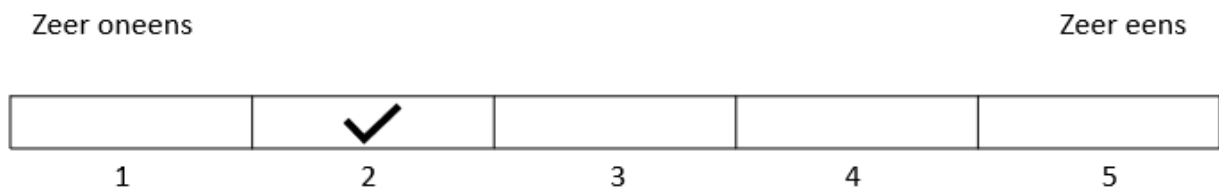
Think aloud methode

Subjectief meten kan op meerdere manieren. Een manier van subjectief meten is het Think Aloud Protocol (denk hardop protocol). Dit is een methode waarbij deelnemers gevraagd worden hun gedachten hardop uit te spreken terwijl ze een set specifieke taken uitvoeren, zoals 'verander de taal van de applicatie' of 'verander je profielfoto'. Hierbij bespreekt de deelnemer zijn exacte handelingen, zoals 'Ik druk op de knop met het camera icoontje' of 'Ik zoek naar de instellingen', maar ook gedachten of meningen zoals 'Ik vind deze knop op een erg onhandige plek zitten'. De onderzoekers verzamelen informatie over hoe de deelnemer de taken uitvoert. Deze informatie wordt gebruikt om de usability te beoordelen en inzicht te krijgen in mogelijke verbeteringen. Meer uitleg over de Think Aloud methode wordt gegeven door de docent tijdens de les. Een andere versie van de think aloud methode is de Retrospective Think Aloud methode. Hierbij vertellen de deelnemers pas na het uitvoeren van de opgedragen taken hun gedachten. Het uitvoeren van deze taken kan worden opgenomen om achteraf met de deelnemers door te nemen. Het voordeel van deze methode is dat de deelnemers zich kunnen focussen op de taken zonder afleidingen. Een nadeel van deze methode is dat deelnemers wellicht handelingen en gedachten vergeten.

System Usability Scale

Een andere optie om subjectief te meten is door het gebruik van de System Usability Scale (SUS). De SUS is een vragenlijst met tien vragen. De deelnemers kunnen deze vragen beantwoorden met één van de vijf volgende opties: zeer oneens, oneens, neutraal, eens of zeer eens. Deze vijf opties worden vervolgens vertaald naar getallen. Deze zie je onder de hokjes.

De SUS is fijn om te gebruiken omdat het erg gemakkelijk is voor de deelnemers. De SUS is daarnaast ook geschikt en betrouwbaar bij een kleine groep deelnemers. Naast de SUS zijn er nog andere voorbeelden van standaardvragenlijsten zoals de User Experience Questionnaire (UEQ) met meer focus op UX.



Figuur 9: De schaal bij SUS.

De tien vragen die aan de deelnemers gevraagd worden zijn deze:

1. Ik denk dat ik dit systeem regelmatig zou willen gebruiken.
2. Ik vond het systeem onnodig complex.
3. Ik vond dat het systeem gemakkelijk te gebruiken was.
4. Ik vind dat ik hulp nodig zou hebben van een technisch persoon om dit systeem te gebruiken.
5. Ik vond dat de verschillende functies in dit systeem goed geïntegreerd waren.
6. Ik vond dat er teveel inconsistentie in dit systeem was.
7. Ik kan me voorstellen dat de meeste mensen snel zullen leren hoe ze dit systeem gebruiken
8. Ik vond het systeem lastig te gebruiken.
9. Ik voelde me zeer zelfverzekerd bij het gebruik van dit systeem.
10. Ik moest veel leren voordat ik met dit systeem aan de slag kon gaan.

Bovenstaande vragen kun je aanpassen aan hetgeen dat je wilt testen. Hier mag dus ook 'feature' of 'aspect' staan in plaats van systeem.

De resultaten uit een System Usability Scale onderzoek kunnen per vraag 1, 2, 3, 4 of 5 zijn. Met deze nummers kun je een SUS score berekenen. Dit is een score tot 100. Deze score is geen percentage, maar geeft aan hoe sterk de usability van jouw systeem is.

Dit zijn de stappen voor het berekenen van een SUS score:

1. Trek voor oneven genummerde vragen 1 van de score af.
2. Trek voor elke even genummerde vraag de score van 5 af.
3. Tel de nieuwe waarden bij elkaar op en vermenigvuldig deze met 2,5.

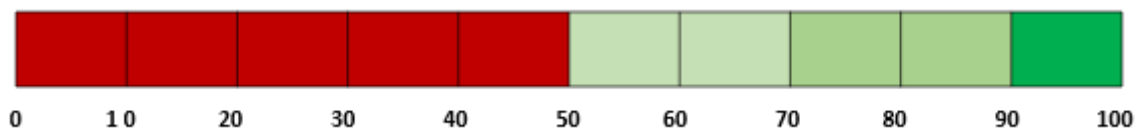
De formule bij stap 1-2 is:

$$(X1 - 1) + (5 - X2) + (X3 - 1) + (5 - X4) + (X5 - 1) + (5 - X6) + (X7 - 1) + (5 - X8) + (X9 - 1) + (5 - X10)$$

Waarbij X1,...,X10 de score is die bij een vraag gegeven is.

Wanneer je de SUS score van de gehele test wilt berekenen herhaal je alle stappen voor alle gebruikers en bereken je de gemiddelde SUS score. Hoe hoger de score van jouw SUS test, des te beter jouw systeem is in usability.

Een SUS score <60 is onvoldoende, 60-70 is voldoende, 70-80 is goed, 80-90 is zeer goed en >90 is uitmuntend. De ultieme score is dus 100. Volgens een onderzoek van Bangor et al. is het gemiddelde is 70.14.



Figuur 10: De schaal van de SUS score.

OPDRACHT 37. THINK ALOUD

Opdracht: Je werkt in 2-tallen digitaliseer de volgende quiz in Microsoft Forms (<https://forms.office.com>). Alle stappen die je neemt spreek je hardop uit. Je buurman maakt hiervan notities (think aloud methode). Vanaf vraag 2 geef je de laptop door en wisselen jullie van rol.

Quiz: Usability – Think Aloud & SUS

Vraag 1

Wat is het belangrijkste kenmerk van de Think Aloud-methode?

(1 goed antwoord)

- A. De gebruiker vult na afloop een vragenlijst in
 - B. De onderzoeker legt alles uit tijdens de test
 - ☒ C. De gebruiker zegt hardop wat hij of zij denkt tijdens het uitvoeren van een taak
 - D. De gebruiker werkt in stilte om zo min mogelijk beïnvloed te worden
-

Vraag 2

Wat meet de System Usability Scale (SUS)?

(1 goed antwoord)

- A. Hoeveel fouten de gebruiker maakt
 - B. Hoe snel iemand een taak afrondt
 - ☒ C. De gebruiksvriendelijkheid van een systeem volgens de gebruiker
 - D. Of het systeem technisch goed werkt
-

Vraag 3

Hoeveel stellingen bevat de System Usability Scale (SUS)?

(1 goed antwoord)

- A. 5
- B. 8
- ☒ C. 10
- D. 12

Think aloud methode

De taken die de deelnemers krijgen:

1. _____
2. _____
3. _____

Noteer hieronder de uitspraken van de deelnemers.

Hadden de deelnemers de juiste denkrichting bij het uitvoeren van de taken? Konden ze dingen goed vinden, of juist niet? Verklaar je antwoord.

OPDRACHT 38. SUS SCORE

Beantwoord nu de vragen horende bij de SUS over Microsoft Forms.

	Zeer oneens Zeer eens				
	1	2	3	4	5
Ik denk dat ik dit systeem regelmatig zou willen gebruiken.					
Ik vond het systeem onnodig complex.					
Ik vond dat het systeem gemakkelijk te gebruiken was.					
Ik vind dat ik hulp nodig zou hebben van een technisch persoon om dit systeem te gebruiken.					
Ik vond dat de verschillende functies in dit systeem goed geïntegreerd waren.					
Ik vond dat er teveel inconsistentie in dit systeem was.					
Ik kan me voorstellen dat de meeste mensen snel zullen leren hoe ze dit systeem gebruiken					
Ik vond het systeem lastig te gebruiken.					
Ik voelde me zeer zelfverzekerd bij het gebruik van dit systeem.					
Ik moest veel leren voordat ik met dit systeem aan de slag kon gaan.					

SUS score

	1	2	3	4	5	6	7	8	9	10
Gebruiker 1										
Gebruiker 2										
Gebruiker 3										

Score gebruiker 1: _____

Score gebruiker 2: _____

Score gebruiker 3: _____

Gemiddelde score: _____

OPDRACHT 39. EVALUATIE

Vul nu op basis van je onderzoek in opdracht 38 en 39 de **Opvallende zaken en verbetervoorstellen** in.

Opvallende zaken

1.

2.

3.

4.

5.

Verbetervoorstellen

1.

2.

3.

OPDRACHT 40. INLEIDENDE OPDRACHT

In groepen van zes tot zeven personen gaan jullie discussiëren over de antwoorden op een vraag die je van de docent krijgt.

Benoem voordat je begint één van de groepsleden tot schrijver. De rol van de schrijver is om de informatie die besproken wordt in de groep op te schrijven.

Het doel van deze opdracht is het simuleren van een focus groep. In deze les leer je wat een focus groep is.

INTERVIEW EN FOCUSGROEP

Vorige les zijn twee subjectieve evaluatie methoden besproken, het Think Aloud protocol en de System Usability Scale (SUS). Deze les leer je twee andere subjectieve methodes kennen, namelijk interviews en focus groepen.

Een interview is een van de meest voorkomende vorm van kwalitatief onderzoek. Een interview is formeel gezegd *'het verzamelen van informatie uit mededelingen van een ondervraagde persoon of personen ter beantwoording van één of meer vooraf bedachte, vaak nog zeer globale vraagstellingen'*. Een interview is dus eigenlijk een gestructureerd gesprek. Er zijn meerdere vormen voor interviews die je kan gebruiken.

1. Het ongestructureerde interview

Het interview bestaat uit een paar open vragen. Deze vragen zijn bedoeld om ruim op te antwoorden.

2. Het semi-gestructureerde interview

Het interview bestaat uit iets meer vragen dan een ongestructureerd interview. De vragen zijn iets scherper maar laten wat ruimte voor eigen inbreng van de geïnterviewden.

3. Het gestructureerde interview

Het interview bevat de meeste vragen en de vragen zijn specifiek.

4. Focusgroep

Een *focusgroep* is een soort groepsinterview om informatie te verkrijgen. Het belang van een focusgroep is om inzicht te krijgen in de meningen van de deelnemers. De groep bestaat uit vijf tot tien deelnemers met specifieke eigenschappen die representatief zijn voor de doelgroep die onderzocht wordt. Voor het testen van een app voor makelaars zouden deze tien deelnemers bijvoorbeeld allemaal makelaars kunnen zijn. Specifieke eigenschappen kunnen ook gaan over afkomst, leeftijd, hobby, etc.

Het gesprek wordt geleid door de interviewer. Deze kun je ook 'gespreksleider' of 'moderator' noemen. In het interview worden vooral open vragen gesteld. Deelnemers mogen ook op andere deelnemers reageren. Op deze manier ontstaat er een discussie. Een focusgroep wordt daarom ook wel een groepsdiscussie genoemd. Het discussiëren binnen een groep helpt met het verkrijgen van diepere inzichten.

Met behulp van een focusgroep kun je ontdekken hoe mensen over een bepaald onderwerp denken. Hierbij is ook van belang wat voor woorden gebruikt worden bij het onderwerp, en welke aspecten belangrijk worden gevonden.

Een focusgroep is niet hetzelfde als een groepsinterview. Een groepsinterview is vaak een makkelijkere manier om snel meerdere interviews uit te voeren. Deelnemers reageren hierbij minder (of niet) op elkaar. Het resultaat hiervan is dat mensen vaak wat meer moeite hebben bij het vormen van hun mening. In een focusgroep geeft een andere deelnemer al een mening, en kan een deelnemer op basis daarvan beslissen in hoeverre hij of zij daarin meegaat of er een andere mening tegenover wil stellen.

Als je een focusgroep organiseert, is het belangrijk dat je goed nadenkt over de volgorde van je vragen. De juiste opbouw helpt om het gesprek rustig op te starten, het onderwerp te verkennen, tot de kern te komen, en het gesprek goed af te sluiten. Er zijn vier soorten vragen die je meestal in deze volgorde gebruikt: openingsvragen, inleidende vragen, hoofdvragen en afrondende vragen.

De eerste vraag is de **openingsvraag**. Deze stel je om het gesprek op gang te brengen en om de deelnemers zich op hun gemak te laten voelen. Iedereen moet deze vraag kunnen beantwoorden, en dat moet snel en makkelijk kunnen. Het is niet de bedoeling om meteen naar een mening of gevoel te vragen. Je vraagt bijvoorbeeld naar een gewoonte of ervaring. Denk aan een vraag als: "Hoe vaak gebruik jij sociale media op een dag?" Zo leer je iets over wie er in de groep zitten, zonder dat het al te persoonlijk of moeilijk is.

Daarna stel je een of meer **inleidende vragen**. Deze zorgen ervoor dat het onderwerp van de focusgroep geïntroduceerd wordt. Deelnemers kunnen hierbij nadenken over wat ze al weten of meegemaakt hebben. Dit soort vragen is niet bedoeld om meteen diep te analyseren, maar vooral om het gesprek op gang te helpen. Bijvoorbeeld: "Wanneer heb jij voor het laatst een app gebruikt die je nog niet kende?" Zo geef je mensen de ruimte om ervaringen te delen die aansluiten bij het onderwerp.

Vervolgens komen de **hoofdvraag of hoofdvragen**. Dit zijn de belangrijkste vragen van de focusgroep. Hier wil je de meeste tijd aan besteden. De deelnemers gaan hier hun meningen, ideeën en ervaringen delen, en reageren op elkaar. Meestal zijn dit twee tot vijf vragen. Dit is het moment waarop je écht waardevolle informatie verzamelt. Bijvoorbeeld: "Wat vind je goed of juist slecht aan het ontwerp van deze website?" Deelnemers gaan nu inhoudelijk in op het onderwerp en leren ook hoe anderen erover denken.

Tot slot stel je **afrondende vragen**. Deze zijn bedoeld om het gesprek netjes af te sluiten en om zeker te weten dat je niets belangrijks hebt gemist. Er zijn drie soorten afrondende vragen. Ten eerste kun je een vraag stellen waarin je mensen laat terugblikken op alles wat besproken is, zoals: "Stel dat je één minuut hebt om iets te zeggen tegen de maker van dit product, wat zou je dan zeggen?" Zo ontdek je wat deelnemers het belangrijkste vinden. Ten tweede kun je als gespreksleider een korte samenvatting geven van wat er besproken is, en dan vragen: "Klopt dit met wat jullie bedoelden? Willen jullie nog iets toevoegen?" Dat is de zogenaamde samenvattende vraag. Tot slot stel je vaak nog een laatste, open vraag zoals: "Is er nog iets dat je had willen zeggen, maar dat nog niet aan bod is gekomen?" Vooral bij de eerste focusgroepen is deze vraag nuttig, omdat het je kan helpen om de volgende bijeenkomst beter voor te bereiden.

Door de vragen in deze volgorde te stellen, zorg je voor een goede opbouw in het gesprek. Zo voelen mensen zich op hun gemak, komen ze tot goede gesprekken, en krijg jij de informatie die je nodig hebt voor je onderzoek.



OPDRACHT 41. FOCUSGROEP CASUS (DOCENT BEGELEIDT)

Beschrijving

Sinds dit jaar staan alle studiewijzers op Learnbeat. We willen kijken in deze focusgroep hoe dat werkt en wat jullie ermee kunnen.

Vragen voor de focusgroep (waar app staat kan je ook lezen website)

Startvragen (opwarmen)

1. Wie gebruikt de app, en waarvoor gebruik je hem meestal?
2. Hoe vaak gebruik je de app?

Usability-specifieke vragen

3. Kun je beschrijven wat je van de eerste keer gebruiken van de app vond?
4. Hoe makkelijk is het om te vinden wat je zoekt (zoals je huiswerk)?
5. Zijn er onderdelen waarvan je niet wist wat ze doen of die verwarrend waren?
6. Heb je ooit iets per ongeluk fout gedaan in de app? Wat gebeurde er toen?
7. Wat vind je van de vormgeving van de app? (lettertype, knoppen, indeling)

Verdieping

8. Als je de app een cijfer zou geven voor gebruiksgemak, welk cijfer zou dat zijn? Kun je dat toelichten?
9. Wat zou je veranderen of verbeteren aan de app?
10. Zijn er apps die je fijner vindt werken? Wat doen die beter?

Afsluiting

11. Wat wil je nog kwijt over je ervaring met de app?
12. Stel dat jij het ontwerp mocht aanpassen: wat zou je als eerste aanpakken?

Focus groep

De vragen aan de deelnemers:

1. _____
2. _____
3. _____

Noteer hieronder de uitspraken van de deelnemers.

Welke conclusies kun je trekken uit de antwoorden op de vragen? Hoe kan daarmee het ontwerp worden verbeterd? Verklaar je antwoord.

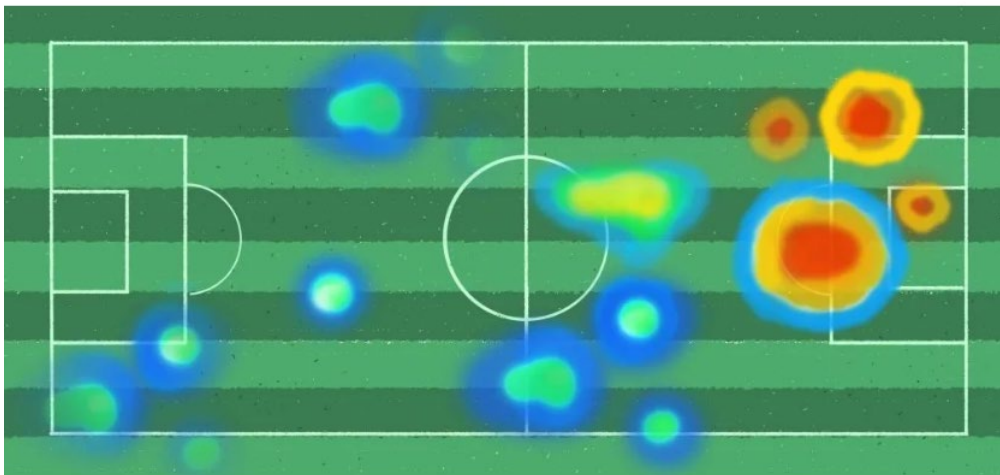
LES 10B OBJECTIEVE EVALUATIE METHODEN

In de vorige les heb je over subjectieve evaluatie methoden geleerd. Deze les richt zich op de objectieve evaluatie methoden. Hierbij ligt de focus op meetbare waarden, zoals tijdsmetingen, mouse tracking, eye tracking, heatmaps en fysiologische metingen. Het testen op meetbare waarden wordt gedaan met tools die hiervoor zijn ontwikkeld.

Heatmap

Een heatmap is een methode waarbij numerieke data grafisch wordt laten zien. Deze data is afkomstig van individuele data punten in een matrix verschillende kleuren voorstellen. De kleuren in de heatmap kunnen verschillende frequenties van een gebeurtenis voorstellen, maar ook een hoeveelheid per locatie.

Een heatmap is te gebruiken voor allerlei doeleinden, bijvoorbeeld inzicht krijgen in het loopgedrag van een voetbal speler.



Afbeelding 13: Heatmap met informatie over waar een bepaalde voetballer het meest loopt.

Mouse tracking

Het opnemen van muisgedrag (ook wel mouse tracking of cursor tracking) kan informatie geven over wat mensen doen op een interface. Metrieken die je hier uit zou kunnen krijgen zijn bijvoorbeeld het aantal clicks, het aantal schermen die mensen gezien hebben, en of er gescrold is tot de pagina grens. Verder kan je ook kijken naar het aantal fouten gemaakt, en het aantal fouten gemaakt ten opzichte van het aantal clicks. Ook de hoeveelheid dat mensen terug klikken kan indiceren dat ze verdwaald zijn of op de verkeerde pagina waren. Tot slot kun je zien of mensen zich aan het ideale pad houden om een taak te bereiken.

Tijdsmetingen

Met behulp van tijdsmetingen kan je testen hoe lang iets duurt, bijvoorbeeld hoe lang gebruikers over een taak doen of hoelang een website nodig heeft om volledig te laden. Daarnaast kun je ook verschillende ontwerpen met elkaar vergelijken en zien bij welk ontwerp gebruikers hun taken het

snelst kunnen uitvoeren. Je zou ook nog tijdsmetingen kunnen programmeren in JavaScript. Hiermee zou je kunnen testen hoe lang het duurt voordat een pagina geladen is.

Bij tijdsmetingen meet je eerst het tijdstip aan het begin, bijvoorbeeld 13:03:42:100 (1 uur, 3 minuten en 41,1 seconden). Vervolgens begin je de test, bijvoorbeeld het laden van een website. Wanneer de website volledig geladen is leg je het tijdstip weer vast, bijvoorbeeld 13:04:59:802.

Het eindtijdstip min het begin tijdsstip levert dan de laadtijd op. In dit voorbeeld is dat 18,7 seconden.



Fysiologische metingen

Met behulp van fysiologische metingen kan je lichamelijke reacties meten. Zo kan je testen of de hartslag van een gebruiker omhoog gaat bij het gebruik van een product of dat er andere tekenen van opwinding zijn, zoals zweten. Andere voorbeelden van fysiologische metingen die gedaan kunnen worden zijn bloeddrukmetingen, ademhaling en spierspanning.

Afbeelding 14: Garmin activity tracker: Een armband die je hartslag meet.

Eye tracking

Eye tracking is de opname van de positie en de beweging van het oog van de gebruiker en de beweging in een bepaalde omgeving om visuele aandacht vast te stellen. Dit kan met behulp van een speciale bril gebaseerd op de optische opname van hoornvlies reflecties.

De resultaten van een eye tracking onderzoek kunnen worden weergegeven in een gaze plot. Een gazeplot laat de locatie, volgorde en totale tijdsbesteding bij het kijken naar een specifieke locatie op een pagina, plaatje of video zien. Hierdoor kunnen onderzoekers achterhalen waar het meest en het langst naar gekeken wordt tijdens bijvoorbeeld een advertentie.

De cirkels op het plaatje hieronder geven de locatie van waar de deelnemer gekeken heeft aan. De cijfers binnen deze cirkels geven de volgorde aan waarin de deelnemer het product bekeken heeft. De totale tijdsbesteding van het kijken wordt aangegeven door de diameter van de cirkels op het plaatje hieronder. Hoe langer een deelnemer kijkt, hoe groter de cirkel.



Afbeelding 15: Gaze plot bij een product. [Bron: Tobiiipro]

Hotjar

Hotjar is een service die aan een website toe kan worden gevoegd. Het doel van Hotjar is het verbeteren van de usability van de website. De gebruikers worden bijvoorbeeld gevraagd om feedback te geven over de site, en op die manier kan deze weer verbeterd worden.

Het in gebruik nemen van Hotjar op je website is redelijk simpel: Ten eerste maak je een account aan op hun website, waarna je wordt geïntroduceerd en rondgeleid op de website. De website oogt goed en aantrekkelijk, en is daarnaast ook gebruiksvriendelijk. De tweede stap is om een stukje code toe te voegen aan de website. Ook hiervoor zijn duidelijke instructies beschikbaar. Kort daarna kan je beginnen met het gebruiken van Hotjar! Een ander programma dat je kan gebruiken voor dezelfde doeleinden als Hotjar is Google Analytics.

Wat kun je doen in Hotjar?

Hotjar biedt functies onderverdeeld in analytics en feedback. De meest interessante functie van Hotjar onder analytics is de heatmap. Hier kun je zien waar de bezoekers van jouw site klikken. Die data wordt vervolgens in een overzichtelijk grafiekje gezet. Recordings zijn vergelijkbaar: hier kun je het muisgedrag van de bezoekers terugkijken. Bij funnels is te zien waar op welk punt de bezoekers de website verlaten. Als laatste kun je bij forms bekijken hoeveel tijd het mensen kost om formulieren op je website in te vullen, en welke velden ze vaker open laten.

De functies bij feedback zijn verschillende vormen van het binnenhalen van de meningen van de bezoekers. Bij incoming is de binnenkomende feedback te zien van de widget die je op je site plaatst. Daarnaast zijn er polls, surveys en een service waarbij gebruikers worden gevraagd om te helpen bij de ontwikkeling van de website, genaamd recruiters.

Er is ook een site dashboard waar een globaal informatie overzicht wordt getoond. Er zijn verschillende abonnementen beschikbaar, afhankelijk wat je wil gebruiken. Voor deze lesmodule maken we gebruik van de gratis versie.

OPDRACHT 42. HOTJAR

Het mooiste is als we ook iets met Hotjar kunnen doen! In deze opdracht voegen we Hotjar toe aan de door jouw ontwikkelde website.

Maak een account en voeg de code op Hotjar toe aan de <head> van de pagina die je wilt testen toe. Het programma moet nu een aantal dagen runnen om resultaten te meten. Je gaat een andere les dus kijken naar de resultaten.

Wanneer je klaar bent kun je alvast mensen uitnodigen om op je site te bekijken. Je hebt dus wel wat bezoekers nodig om resultaten te krijgen in Hotjar.

NPS

Een klanttevredenheidsonderzoek is niet altijd voldoende om erachter te komen hoe een klant over een bedrijf denkt. Een reden waarom is dat enquêtes vaak lang zijn of lang duren. Een klant haakt dan sneller af bij het invullen van dit onderzoek.

Een andere techniek om de mening van de klant te achterhalen is door middel van de Net Promotor Score (NPS). Bij de NPS wordt er aan de klant maar één vraag gesteld, namelijk *'Hoe waarschijnlijk is het dat u dit product zou aanbevelen aan een vriend of collega?'*.

Een klant kan deze vraag beantwoorden met een cijfer van nul tot tien.

Klanten worden na het beantwoorden van deze vraag opgedeeld in drie categorieën.

Categorie 1: Detractors (tegenstanders) – Klanten die een score tussen de nul en zes gaven. Deze klanten zijn hoogstwaarschijnlijk ontevreden over het product.

Categorie 2: Passives (passief tevreden) – Klanten die een score van zeven of acht gaven. Deze klanten zijn positief over het product.

Categorie 3: Promoters – Klanten die een score van negen of tien gaven. Deze klanten zijn zeer enthousiast over het product.

De NPS zelf is niet het cijfer dat de klant het product geeft, maar een score tussen de -100 en 100. Het wordt berekend door het percentage tegenstanders af te trekken van het percentage promoters.

Deze berekening ziet er zo uit:

$$\text{Net Promoter Score} = \% \text{ Promoters} - \% \text{ Tegenstanders}$$

Als het percentage promoters bijvoorbeeld 35 procent is, en het percentage tegenstanders 30 procent, is de NPS vijf.

Hoe hoger de NPS score (tussen -100 en 100), hoe tevredener klanten zijn.

OPDRACHT 43. INVLOED VAN FACTOREN OP NPS

De NPS kan afhangen van andere meespelende factoren zoals bekendheid van het product of verschil in omgeving van de gebruiker. Leg voor beide factoren uit op welke wijze dit verschil kan maken.

Card sorting

Het is voor een gebruiker erg belangrijk dat hij of zij kan vinden waar hij naar zoekt. Wanneer een product, site of app meerdere opties geeft, moet de ontwerper zich afvragen of de informatie op een logische plek staat. Een overzichtelijke *navigatiestructuur* is een belangrijk onderdeel voor een optimale usability. Card sorting (ook wel tree test) is een manier waarmee je dit kan testen. Card sorting is erg simpel en kan zowel online als op papier worden uitgevoerd. Het doel van dit onderzoek is het categoriseren van onderwerpen zodat er een informatiestructuur ontstaat die voldoet aan de verwachtingen van de gebruiker. De gebruiker gaat op zijn intuïtie af om de informatie proberen te vinden

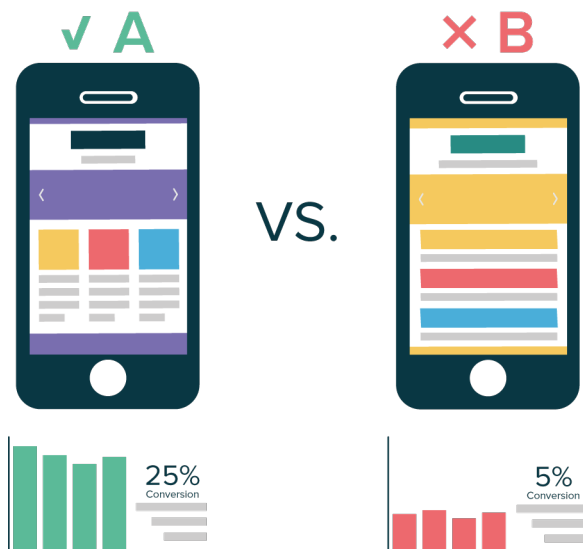
Bij deze test wordt de gebruiker een situatie voorgelegd waarbij hij of zij informatie nodig heeft. Vervolgens wordt de gebruiker gevraagd om aan te geven welke menu's hij of zij zou aanklikken.



Afbeelding 16: Twee verschillende manieren waarop meubels in een menu kunnen gezet worden, op basis van verschillende categorieën. Tijdens een test zou bijvoorbeeld gevraagd kunnen worden of de deelnemer de bedden kan vinden. Wellicht kan de deelnemer beter inschatten hij of zij de pagina met bedden kan vinden in de eerste navigatie structuur, of juist in de tweede navigatie structuur.

A/B testen

Bij een a/b test worden verschillende varianten van een design getest. Gebruikers krijgen willekeurig een van de designs te zien. Door te kijken naar het gebruikers gedrag kan een onderzoeksteam zien welke variant het beste werkt. Gebruikers weten niet dat zij getest worden. Je krijgt hierdoor makkelijk feedback over hoeveel conversie je hebt voor een bepaalde versie. Bijvoorbeeld, hoeveel mensen laten hun e-mail adres achter, hoeveel mensen kopen iets of vullen een formulier in. Je kan de gegevens per versie weergeven in een histogram om gemakkelijk de verschillen te zien. Een nadeel van A/B testen is dat je de reden achter de verschillen niet weet, omdat je niet met de deelnemers spreekt.



Afbeelding 17: Een voorbeeld van A/B testen: A is een ander design dan B. Uit deze test blijkt dat design A de voorkeur heeft bij de gebruikers. [Bron: paulvanderlaken.com]

OPDRACHT 44. CASE STUDY

In deze opdrachten ga je een in tweetallen een navigatiestructuur maken bij één van de onderstaande case studies. De docent zal aangeven welke case study, A of B, jullie zullen gebruiken.

Lees de aangewezen case studie en maak een navigatie structuur bij de informatie uit de tekst. Dit kunnen jullie zowel via Microsoft Word (door een structuur te maken met figuren) doen, of op papier.

Hierna gaan jullie deze structuur testen bij een ander tweetal, die de andere case study hebben gekregen. Dit doe je door de deelnemer te vragen om de taken, gegeven onder de case studie, uit te voeren. Na elke taak noteer je je bevindingen in het antwoordenveld onder de taken.

Case Study A



Het bedrijf USBA verkoopt smartwatches. Dit zijn horloges waarbij je met behulp van een verbinding met je smartphone applicaties kan gebruiken. Deze smartwatches zijn in allerlei varianten te verkrijgen. Een smartwatch kan een kleur (wit, zwart en bruin), een vorm (rond en vierkant) en een andere interface hebben. Deze kan de gebruiker zelf kiezen. Het bedrijf levert de smartwatches zelf.

Afbeelding 18: Voorbeeld van een smart watch. [Bron: Quvoo]

Er zijn drie verschillende interfaces beschikbaar en iedereen heeft zijn eigen voordelen en nadelen. Een gebruiker kan ook wisselen tussen interfaces, maar hier zitten kosten aan verbonden. Een gebruiker betaalt tien euro voor elke keer dat hij of zij een interface wil veranderen.

- **Interface A** maakt gebruik van een swipe optie. Op het scherm wordt maar één applicatie tegelijk getoond. Een gebruiker kan 4 keer swipen om een andere applicatie te zien.
- **Interface B** maakt gebruik van een home screen. Op het scherm zijn vier applicaties te zien. Een gebruiker kan op een applicatie klikken.
- **Interface C** maakt gebruik van een wiel op het scherm. Door het wiel te draaien kan een gebruiker een andere applicatie kiezen.

De interfaces zijn te zien in de afbeelding onder de case studie.

Om een smartwatch te bestellen moet een gebruiker een account maken. Op dit account kun je je vorige bestellingen, je totale uitgaven en je huidige interface zien.

De website van USBA biedt ook hulp aan bij het kiezen van interfaces, hun retourneer policy en hulp bij kapotte smartwatches.

Tot slot biedt USBA ook opties voor bedrijven om samenwerkingen aan te gaan.



Figuur 10: De interfaces van de smartwatch

Tijdens de test zal je de deelnemer de volgende taken geven:

Navigeer naar het kopje waar jij denkt dat de informatie die je nodig hebt zal zijn.

- Je wilt van interface veranderen.
- Je wilt de kosten van het veranderen van de interface zien.
- Je wilt een smartwatch kopen maar vraagt je af hoeveel het kost om dit te laten bezorgen.
- Je wilt je adres in je account aanpassen.
- Je wilt je laten adviseren over het interface dat je zou moeten kiezen.
- Je wilt graag een project starten met USBA.

Vragen voor leerlingen met case study A:

Noteer onder de taak je bevindingen. Waar zocht de deelnemer het eerst? Was dat correct?

Je wilt van interface veranderen.

Je wilt de kosten van het veranderen van de interface zien.

Je wilt een smartwatch kopen maar vraagt je af hoeveel het kost om dit te laten bezorgen.

Je wilt je adres in je account aanpassen.

Je wilt je laten adviseren over het interface dat je zou moeten kiezen.

Je wilt graag een project starten met USBA.

Kon het andere team de juiste locatie voor alle informatie vinden? Verklaar je antwoord.

Wat zou je verbeteren in je navigatie structuur op basis van de resultaten uit de test? Verklaar je antwoord.

Case study B



Het bedrijf Light4You verkoopt smart alarms. Smart alarms zijn wekkers met lampen die van kleur kunnen veranderen en een display. De smart alarm kan bediend worden met een applicatie. Een smart alarm heeft meerdere functies, afhankelijk van welk abonnement de gebruiker heeft. Een gebruiker kan wisselen van abonnement na aanschaf van een smart alarm. Hiervoor betaalt de klant alleen de extra kosten van de upgrade, of in een geval van een downgrade, minder. De smart alarm wordt bezorgd door het bedrijf zelf.

Afbeelding 19: Voorbeeld van een smart light. [Bron: TFA Dostmann]

De verschillende abonnementen en bijbehorende functies staan hieronder uitgelegd:

Abonnement Basic (geen kosten naast de aanschaf van het product):

- De kleuren van de lamp kunnen veranderd worden in de applicatie.
- De wekker kan gezet worden in de applicatie.
- Het display laat de tijd, de temperatuur (buiten) en de datum zien.

Abonnement Comfy (2 euro per maand naast de aanschaf van het product):

- De kleuren van de lamp kunnen veranderd worden in de applicatie.
- De wekker kan gezet worden in de applicatie.
- Het display laat de tijd, de temperatuur (buiten) en de datum zien.
- Het display maakt een connectie met je telefoon en kan via Bluetooth muziek afspelen.
- Het display laat de meest recente nieuwsberichten zien.

Abonnement Pro (3 euro per maand naast de aanschaf van het product)

- De kleuren van de lamp kunnen veranderd worden in de applicatie.
- De wekker kan gezet worden in de applicatie.
- Het display laat de tijd, de temperatuur (buiten) en de datum zien.
- Het display maakt een connectie met je telefoon en kan via Bluetooth muziek afspelen.
- Het display laat de meest recente nieuwsberichten zien.
- Het display laat Whatsapp en Messenger berichten zien.
- Het display laat je agenda afspraken zien.

Om een smart alarm te bestellen moet je een account maken in de app. Op dit account kun je je licht en alarm instellingen zien, en in het geval van een Comfy en Pro abonnement de connectie met je telefoon beheren.

De applicatie voor de smart alarm bevat ook informatie over de privacy rechten van de gebruiker. Daarnaast biedt de applicatie van Light4You ook informatie over hun retour policy en een Q&A pagina voor veelgestelde vragen.

Tijdens de test zal je de deelnemer de volgende taken geven:

Navigeer naar het kopje waar jij denkt dat de informatie die je nodig hebt zal zijn.

- Je wilt van abonnement veranderen
- Je wilt de kosten van het veranderen van het abonnement zien.
- Je wilt een smart alarm kopen maar vraagt je af hoeveel het kost om dit te laten bezorgen.

- Je wilt je adres in je account aanpassen.
- Je wilt je wekker zetten.
- Je wilt je agenda met je telefoon synchroniseren.

Vragen voor leerlingen met case study B:

Noteer onder de taak je bevindingen. Waar zocht de deelnemer het eerst? Was dat correct?

Je wilt van abonnement veranderen.

Je wilt de kosten van het veranderen van het abonnement zien.

Je wilt een smart alarm kopen maar vraagt je af hoeveel het kost om dit te laten bezorgen.

Je wilt je adres in je account aanpassen.

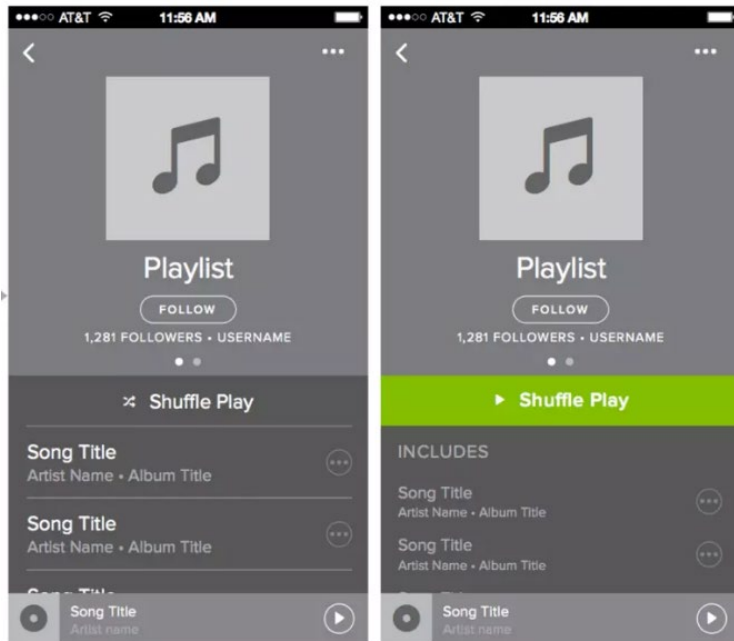
Je wilt je wekker zetten.

Je wilt je agenda met je telefoon synchroniseren.

Kon het andere team de juiste locatie voor alle informatie vinden? Verklaar je antwoord.

Wat zou je verbeteren in je navigatie structuur op basis van de resultaten uit de test? Verklaar je antwoord.

OPDRACHT 45. A/B TESTEN



Afbeelding 20: Spotify A/B test voorbeeld

Bekijk de 2 ontwerpen van de Spotify app en beschrijf welk ontwerp je beter vindt en waarom.

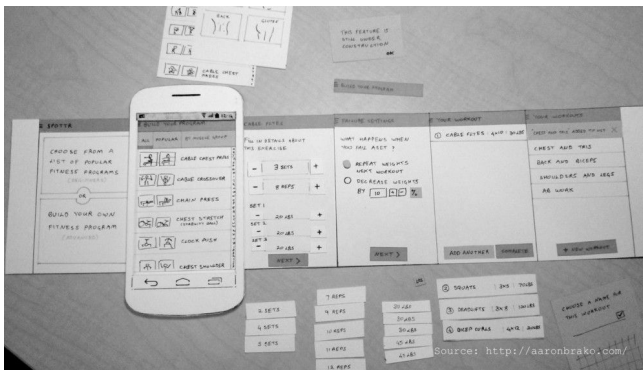
OPDRACHT 46. A/B TESTEN VOORBEREIDEN

In deze opdracht werk je in groepen van vier.

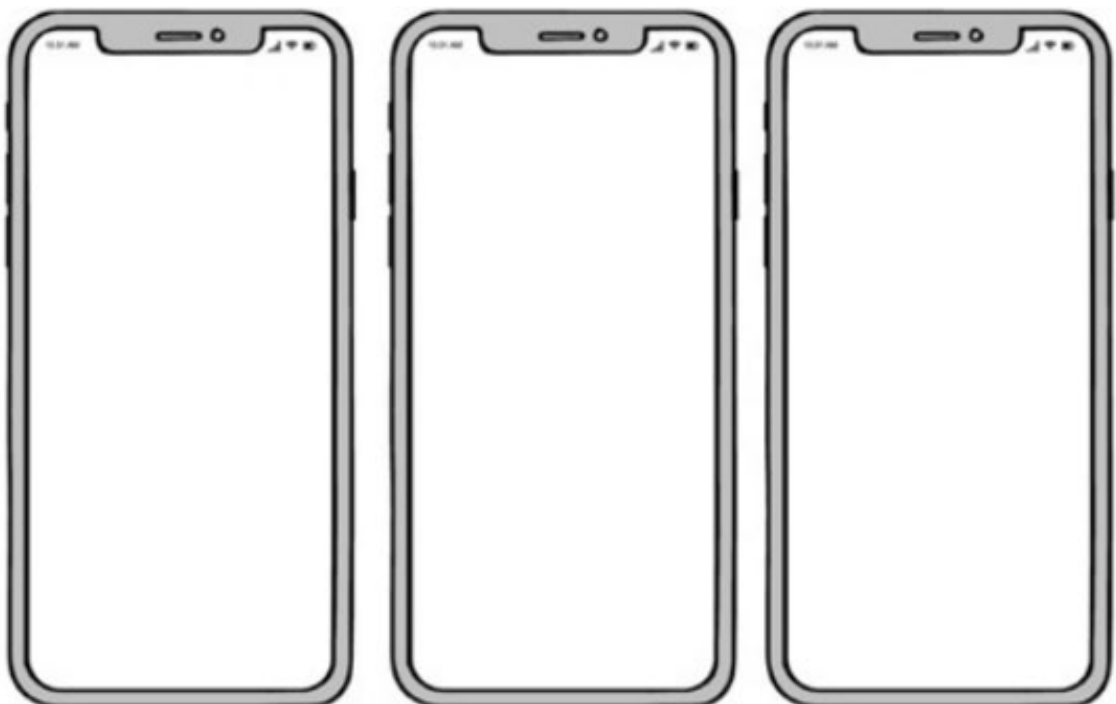
Tijdens deze opdracht ontwerpen jullie een simpele applicatie. Deze applicatie is een educatie app voor jongere kinderen. Op deze applicatie kunnen zij fruit opzoeken en hier informatie over vinden. Deze informatie moet onder andere:

- Een foto van het fruit bevatten;
- Een duidelijk overzicht geven van de maanden waarin de vrucht groeit en andere waardevolle informatie in tekstvorm.

De applicatie mag bestaan uit **maximaal drie schermen** en wordt getekend **op papier** met behulp van de mock-ups op de volgende pagina. Zorg ervoor dat je potlood gebruikt zodat je eventuele fouten kan uitwissen. Ontwerp de app in tweetallen zodat je 2 ontwerpen krijgt. (Let op: je krijgt nu twee heel verschillende ontwerpen. In de praktijk is een A/B test vaak maar een kleine wijziging (zie opdracht 45)).



Afbeelding 12: Een voorbeeld van een op papier ontworpen applicatie met zes schermen.



OPDRACHT 47. THINK ALOUD

In deze opdracht wordt één tweetal tester en het andere tweetal de ontwerpers van de app. De testers gaan de zojuist door hun ontworpen applicatie testen op usability. De ontwerpers geven 1 of 2 opdrachten aan de testers. De test mag volgens uitgevoerd worden op basis van de Think Aloud methode. Noteer je resultaten hieronder. Draai daarna de rollen om en test de andere app.

Think aloud methode

De taken die de deelnemers krijgen:

1. _____
2. _____
3. _____

Noteer hieronder de uitspraken van de deelnemers.

Hadden de deelnemers de juiste denkrichting bij het uitvoeren van de taken? Konden ze dingen goed vinden, of juist niet? Verklaar je antwoord.

OPDRACHT 48. A/B TESTEN

In deze opdracht hebben we de A/B versies besproken in twee think aloud sessies. Je kan ook andere onderzoeksmethodes gebruiken om de twee ontwerpen te beoordelen.

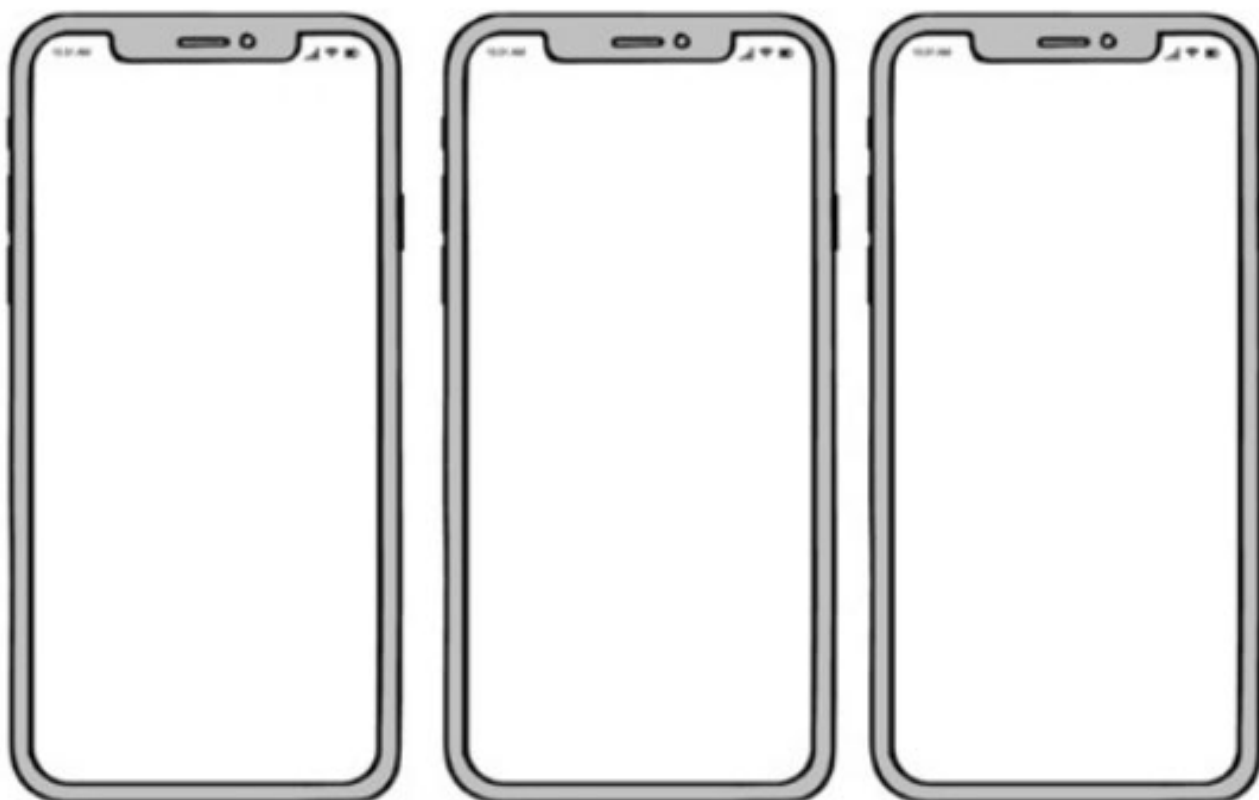
Schrijf hieronder de sterke punten op van beide ontwerpen

Ontwerp 1	Ontwerp 2

Wat zijn de verbeterpunten?

Ontwerp 1	Ontwerp 2

Ontwerp nu één pagina opnieuw op basis van de feedback:



ONDERZOEK UITVOEREN

De volgende twee opdrachten neem je op in een apart document en lever je in in SomToDay.

OPDRACHT 49. ONDERZOEK 1

Kijk naar de opdrachten 3 en 10 op maken.mijn-in.nl. Welke onderzoeksmethode beschreven in hoofdstuk 10 zou je kunnen gebruiken om een antwoord op je onderzoeksvraag te krijgen?

Voer dit onderzoek uit. Kijk daarbij ook naar de volgende opdracht. Het mooiste is als je een subjectieve en objectieve methode hebt gebruikt in beide opdrachten.

OPDRACHT 50. ONDERZOEK 2

Je hebt in hoofdstuk 9 een prototype gemaakt van je website. Voer ook op dit ontwerp een onderzoeksmethode.

INDEX

HTML/ CSS opdrachten

 Opdracht 11.	Je eerste webpagina	27
 Opdracht 12.	Maak je eigen profielpagina	30
 Opdracht 13.	Werken met blokken en inline elementen	32
 Opdracht 14.	Website met structuur	33
 Opdracht 15.	Conventies	36
 Opdracht 16.	Geef je pagina kleur en stijl	39
 Opdracht 17.	Werken met classes	41
 Opdracht 18.	Bouw je eigen gekleurde doos	44
 Opdracht 19.	Tussentoetsje basiskennis html & css	45
 Opdracht 20.	Drie blokken naast elkaar	49
 Opdracht 21.	Flexbox Froggy	50
 Opdracht 22.	Pagina-indeling met Flexbox	50
 Opdracht 23.	Maak je website responsive met media queries	52
 Opdracht 35.	Website volgens functioneel ontwerp	71

Praktische opdrachten onderzoek

 Opdracht 3.	Start praktische opdracht onderzoek	9
 Opdracht 5.	UX Aspecten	12
 Opdracht 10.	Onderzoek starten	25
 Opdracht 26.	Methodes	58
 Opdracht 49.	Onderzoek 1	101
 Opdracht 50.	Onderzoek 2	101

Opdrachten functioneel ontwerp

	Opdracht 27.	Functioneel ontwerp start.....	60
	Opdracht 28.	doel en doelgroep.....	61
	Opdracht 29.	Eisen en wensen	62
	Opdracht 30.	User stories.....	64
	Opdracht 31.	Sitemap	65
	Opdracht 32.	Basiselementen	66
	Opdracht 33.	Schermontwerp	69
	Opdracht 34.	Afronding van het functioneel ontwerp.....	69

Overige opdrachten

	Opdracht 1.	Kennismakingsopdracht.....	6
	Opdracht 2.	Yahoo!	8
	Opdracht 4.	Quiz	12
	Opdracht 6.	Zelf regels bedenken *	12
	Opdracht 7.	Aspecten plaatsen.....	22
	Opdracht 8.	Richtlijnen plaatsen.....	23
	Opdracht 9.	Verbeteren vanuit heuristieken.....	24
	Opdracht 24.	Terugblik eerste lessen	55
	Opdracht 25.	Methodes zoeken	56
	Opdracht 36.	Huiswerk testobjecten.....	74
	Opdracht 37.	think aloud.....	77
	Opdracht 38.	SUS Score	79
	Opdracht 39.	evaluatie	80
	Opdracht 40.	Inleidende opdracht	81
	Opdracht 41.	Focusgroep casus (Docent begeleidt).....	83
	Opdracht 42.	Hotjar	88
	Opdracht 43.	Invloed van factoren op NPS	89
	Opdracht 44.	Case study.....	91
	Opdracht 45.	A/B testen	96
	Opdracht 46.	A/B Testen voorbereiden	97
	Opdracht 47.	Think aloud	98
	Opdracht 48.	A/B Testen	99

Opdrachten per les

Inhoudsopgave	1
Inleiding.....	2
Leerdoelen.....	3
Vooraf.....	4
 Opdracht 1. Kennismakingsopdracht	6
Les 1 Introductie Usability.....	7
 Opdracht 2. Yahoo!.....	8
 Opdracht 3. Start praktische opdracht onderzoek.....	9
 Opdracht 4. Quiz.....	12
 Opdracht 5. UX Aspecten	12
 Opdracht 6. Zelf regels bedenken *.....	12
 Huiswerk leesopdracht richtlijnen	13
Les 2 Heuristieken	20
 Opdracht 7. Aspecten plaatsen	22
 Opdracht 8. Richtlijnen plaatsen	23
 Opdracht 9. Verbeteren vanuit heuristieken	24
 Opdracht 10. Onderzoek starten	25
Les 3 Introductie HTML	26
 Opdracht 11. Je eerste webpagina	27
 Opdracht 12. Maak je eigen profielpagina	30
 Opdracht 13. Werken met blokken en inline elementen	32
 Opdracht 14. Website met structuur.....	33
 Opdracht 15. Conventies	36
Les 4 Introductie CSS	37
 Opdracht 16. Geef je pagina kleur en stijl.....	39
 Opdracht 17. Werken met classes	41
 Opdracht 18. Bouw je eigen gekleurde doos.....	44
 Opdracht 19. Tussentoetsje basiskennis html & css.....	45
Les 5 Flexbox	48
 Opdracht 20. Drie blokken naast elkaar	49
 Opdracht 21. Flexbox Froggy	50
 Opdracht 22. Pagina-indeling met Flexbox.....	50
Les 6 Responsive design	51

Opdracht 23.	Maak je website responsive met media queries	52
Les 7 Ontwerp cyclus.....		54
Opdracht 24.	Terugblik eerste lessen	55
Opdracht 25.	Methodes zoeken	56
Opdracht 26.	Methodes.....	58
Les 8 Functioneel ontwerp		59
Opdracht 27.	Functioneel ontwerp start	60
Opdracht 28.	doel en doelgroep.....	61
Opdracht 29.	Eisen en wensen	62
Opdracht 30.	User stories	64
Opdracht 31.	Sitemap	65
Opdracht 32.	Basiselementen.....	66
Opdracht 33.	Schermontwerp	69
Opdracht 34.	Afronding van het functioneel ontwerp	69
Les 9 Prototype maken.....		71
Opdracht 35.	Website volgens functioneel ontwerp.....	71
Les 10 Evaluatie methodes.....		73
Opdracht 36.	Huiswerk testobjecten	74
Les 10a Subjectieve evaluatie methoden.....		75
Opdracht 37.	think aloud	77
Opdracht 38.	SUS Score	79
Opdracht 39.	evaluatie.....	80
Opdracht 40.	Inleidende opdracht.....	81
Opdracht 41.	Focusgroep casus (Docent begeleidt)	83
Les 10b Objectieve evaluatie methoden.....		85
Opdracht 42.	Hotjar	88
Opdracht 43.	Invloed van factoren op NPS.....	89
Opdracht 44.	Case study	91
Opdracht 45.	A/B testen	96
Opdracht 46.	A/B Testen voorbereiden.....	97
Opdracht 47.	Think aloud.....	98
Opdracht 48.	A/B Testen.....	99
Onderzoek uitvoeren.....		101
Opdracht 49.	Onderzoek 1.....	101

 Opdracht 50. Onderzoek 2	101
Index	102
Extra html/ css - Eenheden.....	107
Logboek	109

EXTRA HTML/ CSS - EENHEDEN

In CSS geef je vaak **afmetingen** op, bijvoorbeeld voor:

- lettergroottes (font-size)
- afstanden (margin, padding)
- breedtes en hoogtes (width, height)

Dat doe je met een **getal + een eenheid**. Bijvoorbeeld: 20px, 50%, 2em.

Hieronder zie je de **meest gebruikte eenheden**.

1. Pixels (px)

- Vaste maat
- 1 pixel = 1 schermpunt (ongeveer)
- Wordt veel gebruikt voor tekst, marges, randen

Voorbeeld:

```
p {  
  font-size: 16px;  
  padding: 10px;  
}
```

Voorspelbaar, maar **niet schaalbaar** op kleine schermen

2. Procenten (%)

- Relatief ten opzichte van het **ouder-element**
- Handig voor flexibele layouts

Voorbeeld:

```
div {  
  width: 50%;  
}
```

Schaalbaar, vooral nuttig in Flexbox en Grid

3. em

- Relatief ten opzichte van de **lettergrootte van het element**
- 1em = 100%

```
p {  
  font-size: 1.5em; /* 1,5 keer zo groot als standaardtekst */  
}
```

Goed voor **tekst die meegroeit**

Let op: em kan **stapelen** (nesten → groter dan je denkt)

4. rem (root em)

- Relatief aan de **basislettergrootte van de hele pagina** (meestal 16px)

Voorbeeld:

```
h1 {  
  font-size: 2rem; /* 2 × 16px = 32px */  
}
```

Voorspelbaarder dan em, handig voor consistente schaal

5. vh en vw (viewport height/width)

- Relatief aan het **schermformaat**:
 - 1vh = 1% van de hoogte van het scherm
 - 1vw = 1% van de breedte van het scherm

Voorbeeld:

```
section {  
  height: 100vh; /* Volledige schermhoogte */  
}
```

Handig voor fullscreen-elementen of responsive design

Samenvattend

Eenheid	Betekenis	Type
px	Pixels (vast)	Absoluut
%	Percentage van ouder-element	Relatief
em	Relatief aan eigen tekstgrootte	Relatief
rem	Relatief aan root (meestal 16px)	Relatief
vh/vw	Relatief aan hoogte/breedte van scherm	Relatief

LOGBOEK

Het volgende logboek houdt je individueel bij.

Week	Gewerkt aan	Noem iets wat deze week goed ging of waar je tegenaan liep.

[illegible]

[illegible]

