

Datapunt 2E - CHECK IT!

Maarten Borninkhof

Klas 1A

10-1-2025

Leeruitkomsten en beoordelingscriteria

LUK 2.1

De student verkent diverse oplossingsrichtingen die beantwoorden aan de door de student zelf geformuleerde ontwerpvraag en gebruikt daarbij verschillende aangereikte technieken én technieken om te testen met peers.

- **BC 2.1.1** Je formuleert ontwerp vragen op basis van de theorie bij een aangereikt probleem. [Onderzoeken]

LUK 2.2

De student verbeeldt diverse oplossingsmogelijkheden die voldoen aan de ontwerp eisen en theorie en gebruikt daarbij verschillende aangereikte creatieve technieken en passende prototyping tools om te komen tot HTML/CSS prototypes.

- **BC 2.2.2** Je codeert een interactieve en navigeerbare website, waarmee je jouw HTML / CSS kennis demonstreert. [Ontwerpen]

LUK 2.3

De student laat zien dat ze haar sterktes en zwaktes met betrekking tot 'samen werken' en 'samen experimenteren' kent. Ze verwerkt deze inzichten in een plan dat bestaat uit persoonlijke leerdoelen en concrete acties.

- **BC 2.3.2** Je formuleert heldere leerdoelen die je tijdens de volgende cursussen in de praktijk kunt brengen. [Ontwikkelen]

BC2.1.1 Je formuleert ontwerp vragen op basis van de theorie bij een aangereikt probleem. [Onderzoeken]

Bij het formuleren van de ontwerp vragen heb ik me aan een bepaalde theorie gehouden om te zorgen dat deze vragen goed zijn om mee te werken. De theorie volgt:

HOE STEL JE EEN ONTWERPVRAAG OP?

- ♦ Begin de vraag met "Hoe kun je..."
- ♦ Beschrijf in de vraag heel concreet en op een uitdagende manier wat je met de nieuwe ideeën wilt bereiken.

WAT ZIJN KENMERKEN VAN EEN ONTWERPVRAAG?

- ♦ **Concreet.** Zorg dat de vraag de kern raakt van je probleem.
 - ♦ **Open.** Zorg dat de vraag zo geformuleerd is dat je meteen in ideeën gaat denken. Voorkom dat de vraag al een bepaalde richting aangeeft.
 - ♦ **Positief.** Zorg dat je geen ontkenning gebruikt in de vraag.
 - ♦ **Relevant.** Zorg dat de vraag nut heeft. Bedenk of het inderdaad zo is dat er ideeën moeten komen voor de vraag.
 - ♦ **Kort.** Zorg dat de vraag niet langer is dan één zin. Gebruik makkelijke woorden en geen afkortingen.
-

Bron: www.ontwerpenindeklas.nl

Ik wijs hieronder terug op 3 ontwerp vragen van vorige datapunten. Hiermee wil ik aantonen hoe deze ontwerp vragen juist zijn volgens de theorie, of hoe ze onjuist zijn, en hoe ze beter zouden kunnen opgesteld worden.

Hoe kun je de combinatie van Google Maps & PayPal zo uitgebreid maar gebruiksvriendelijk mogelijk maken? (2B)

- + De vraag is opengesteld. Je kunt hierdoor een grote hoeveelheid antwoorden bedenken op de ontwerp vraag.
- + De vraag gebruikt geen ontkenning.
- + De vraag is relevant. We willen hieruit immers veel nieuwe ideeën ophalen om toe te voegen aan de gecombineerde app.
- + De vraag is opgesteld uit één zin, gebruikt makkelijke woorden en geen afkortingen.

Maar de vraag kan beter...

Er is niet duidelijk aangegeven wat precies de kern van het probleem is. Om deze vraag beter op te stellen zou het bijvoorbeeld zo gesteld kunnen worden:

✚ *Hoe kun je de combinatie van Google Maps & PayPal vullen met functies die spreken tot de gebruiker van de app?*

Nu is de **kern** geraakt van het probleem, namelijk dat **we functies zoeken die spreken tot de gebruiker van de app**. Ook is bekend wie de *doelgroep* is, namelijk *de gebruikers van de app*. Ten slotte is er toegevoegd wat we willen bereiken, namelijk het vullen van de app met functies die spreken tot de gebruiker van de app.

✚ *Hoe kun je mensen in het openbaar het beste benaderen? (2A)*

- + De vraag is concreet en raakt de kern van het probleem aan (het benaderen van mensen in het openbaar).
- + De vraag is open gesteld.
- + De vraag gebruikt geen ontkenning.
- + De vraag is kort en gebruikt geen afkortingen.
- + De vraag geeft een doelgroep aan (mensen in het openbaar).

De vraag op zichzelf is best goed opgesteld, alleen mist er wel wat we met de nieuwe ideeën willen bereiken. Om het doel toe te passen (wat voor 2A was om mensen mee te laten doen aan een samenwerkend spel) stel ik de vraag als volgt op:

✚ *Hoe kun je mensen in het openbaar het beste benaderen om een spel aan te gaan?*

Nu heb ik het doel van de ontwerp vraag toegevoegd, om de intentie van de vraag en de ideeën die eruit komen beter op het totale plaatje te koppelen. Ook stuurt de vraag nog niet in een bepaalde richting, dus de vraag blijft open.

✚ *Hoe categoriseer ik een groot aantal werken zorgvuldig? (2D)*

- + De vraag is concreet en raakt de kern aan van het probleem (veel werken zorgvuldig kunnen categoriseren).
- + De vraag is open gesteld.
- + De vraag gebruikt geen ontkenning.
- + De vraag is kort en gebruikt geen afkortingen.

Of de vraag relevant was op dat moment, dat weet ik niet helemaal zeker. Ik moest nog best wat aan mijn website bouwen en de vraag is **misschien iets te specifiek**. Ook is de vraag opgesteld met een **minder dan makkelijk woord**. Wat bedoel ik dan precies met ‘categoriseren’?

Mijn conclusie voor deze ontwerpvraag is dat ik deze op het verkeerde moment heb gesteld. Deze is beter voor een latere stage in het ontwerp van mijn website.

Om het te verbeteren, kwam ik op de volgende vraag:

Hoe kan ik de grote hoeveelheid werken indelen?

Door deze vraag te stellen is de vraag **makkelijk verwoord**. Ook voorkom ik meer dat ik de vraag al in een bepaalde richting stuur.

BC2.2.2 Je codeert een interactieve en navigeerbare website, waarmee je jouw HTML / CSS kennis demonstreert. [Ontwerpen]

Voorkennis

Ik heb al eens een paar jaar terug geëxperimenteerd met het bouwen van een [website](https://maartenman123.github.io/SBCC/) (https://maartenman123.github.io/SBCC/). Dit deed ik om een oplossing te vinden voor een specifiek probleem in een videogame – er kwam een systeem tekort voor het bijhouden van ‘collectibles’ door de wereld heen – en met dit probleem ben ik begonnen met bouwen. Hierdoor ben ik bekend met een basis van HTML, CSS, maar ook een stukje JavaScript. Ook was een van mijn hulpbronnen w3schools, wat voor deze cursus ook een betrouwbare en hulpzame bron was.

De link naar mijn webpagina is: <https://2152347.portfolio.aimsites.nl/>

Tewerk tijdens de cursus

Tijdens de cursus ben ik begonnen met het maken van de zelfstudie. Hier heb ik, net als de rest van de cursus, de volgende bronnen gebruikt om mijn kennis op te doen:

- *HTML & CSS: design and build websites* van JON DUCKETT
- *W3schools*
- *Stackoverflow* (voor specifiekere oplossingen)

Aangezien het af te raden is om teveel op internet rond te zoeken voor oplossingen (zo weet je niet zeker of je wel HTML & CSS leert schrijven zoals het aangeboden wordt), heb ik extra aandacht besteed aan het begrijpen van de code omtrent een oplossing. Ik kopieer het niet zomaar, ik duik er in en stel mezelf de vragen als ‘begrijp ik deze code?’, ‘hoe kan ik dit de volgende keer zelf schrijven zonder op te zoeken?’ en ‘komt de opbouw van de code overeen met hoe ik het geleerd heb deze cursus?’. Door dit aan te hebben gehouden, verminderde ik de kans om ‘foutief’ mezelf te leren coderen, maar kon ik toch leren van wat ik precies eruit wilde halen. Ook kan ik zeggen dat dit zelden voorkwam, en dat als ik het opzocht het eigenlijk al geschreven kon worden zoals ik het zelf al probeerde; het diende meer als een soort check en vergelijking met mijn eigen geschreven code. Overigens heb ik wel eerst altijd het boek en w3schools geraadpleegd.

Mijn teksteditorkeuze en de voor- en nadelen

Tijdens het begin van de cursus heb ik kennis gemaakt met verschillende teksteditors. Aangezien ik voor mijn eerdere project al Visual Studio Code heb gebruikt was ik hier al meer mee bekend, maar ik ging alsnog testen om te kijken waarom deze teksteditor nou specifiek fijn en bruikbaar voor mezelf was.

	Visual Studio Code	Notepad	Adobe Dreamweaver	Notepad++
Voordelen	gratis, sluit automatisch <code></></code> in, geeft mogelijkheden om makkelijk plugins toe te voegen	gratis, komt al standaard geïnstalleerd met Windows,	Ingebouwd venster waar je in real time je aanpassingen kunt bekijken	Gratis, open-source
Nadelen	iets te veel opties voor mijn liking; het scherm lijkt cluttered	geen gekleurde code > minder overzichtelijk	niet gratis > Adobe CC abonnement gebonden	sluit niet automatisch <code></p></code> in

Mijn uiteindelijke keuze om in door te werken was Visual Studio Code geworden. Toch vond ik niet per se een groot verschil in de verschillende teksteditors. Ik zie mijzelf in de toekomst hier en daar nog wel eens switchen – voor specifieke workflow binnen een team, of om gewoon voor de lol en nieuwsgierigheid eens wat anders te proberen – dus ik bind me niet aan Visual Studio Code vast.

Experimenteren: ‘little experiments’

Om mijn kennis en nieuwsgierigheid verder te testen, heb ik een folder gemaakt waar ik simpelweg kleine websites heb gebouwd om specifieke dingen te testen, voordat ik deze (potentieel) toe zou kunnen voegen aan mijn eigen website.

changing logo	7-12-2024 20:53	Bestandsmap
CSS - gradient	12-12-2024 09:41	Bestandsmap
overflow - worksSection	7-12-2024 23:01	Bestandsmap

Momenteel heb ik dit driemaal gedaan. Mijn plan voor het vervolg is om **deze map aan te blijven vullen met andere experimenten, waardoor ik een heus ‘arsenaal’ op kan bouwen en aan kan tonen dat ik dieper op HTML & CSS in durf te gaan.**

De volgende manieren geef ik aan dat ik een website heb gecodeerd waaraan je kunt zien dat ik kennis over HTML & CSS demonsteer:

HTML:

- Ik heb de juiste basis toegevoegd van waar een HTML-bestand aan moet voldoen.

```
<!DOCTYPE html>
<html lang="en">
```

- Ik weet welke tags nodig zijn om content te weergeven in de browser.

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

  <body>
>   <main> ...
  </main>
</body>

</html>
```

- Ik weet wat er in de <head>-tag thuisheert en weet hoe ik een titel, metadata voor de website kan toevoegen en ik kan stylesheets en fonts linken.

```
<head>
  <title>MRTN BRNNKHF</title>
  <meta charset="UTF-8">
  <meta name="keywords" content="Maarten Borninkhof">
  <meta name="description" content="Maarten Borninkhof's Portfolio Pagina">
  <meta name="author" content="Maarten Borninkhof">
  <!-- Dit voegt metadata toe wat door zoekmachines gebruikt kan worden om de website beter te kunnen opzoeken. -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="css/css-reset.css" type="text/css" rel="stylesheet">
  <link href="css/style.css" type="text/css" rel="stylesheet">
  <!-- Deze linken naar een CSS-stylesheet om te gebruiken op deze pagina. -->
  <link rel="icon" type="image/x-icon" href="images/favicon.png">
  <!-- Voor het toevoegen van een klein icoontje (favicon) aan het tabblad. -->

  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Doto:wght@100..900&family=Geist+Mono:wght@100..900&display=swap" rel="stylesheet">
  <!-- Deze links halen een font van Google Fonts op om te gebruiken op de pagina. -->
</head>
```

- Ik kan comments toevoegen om de code te beschrijven, voor het geval als ik of iemand anders vergeten is wat de code precies hoort te doen.

Zie de afbeelding hierboven, de comments zijn in het groen aangegeven.

- Ik kan html-elementen en -code 'nesten' om overzicht te behouden in het document kan hiermee de link leggen tussen parent-child elementen.

```
<body>
  <main>
    <header class="header" id="top">
      
      <div class="logo">
        <h1>maarten<br>borninkhof</h1>
        <p>&#10022; dancer, designer, more. &#10022;</p>
        <!-- &#10022 is hier een code die een specifieke HTML entity betreft, waar in dit geval een sterretje te zien is. -->
      </div>
    </header>
```

- Ik kan HTML entities toevoegen.

```
<p>&#10022; dancer, designer, more. &#10022;</p>
<!-- &#10022 is hier een code die een specifieke HTML entity betreft, waar in dit geval een sterretje te zien is. -->
```

- Ik kan een -tag toevoegen, deze linken aan een afbeelding, en een alt-beschrijving toevoegen (dat is handig voor als de website de afbeelding niet wilt laden).

```

```

- Ik kan een id- en class-element toevoegen om deze specifieker vorm te geven met CSS.

```
<header class="header" id="top">
<div class="youtubeLinks">
<section id="works">...
</section>
<section id="impressions">...
</section>
<section id="about">...
</section>
<section id="contact">...
</section>
```

Hiermee toon ik ook aan dat ik de class- en id-elementen zorgvuldig kan voorzien van passende benamingen die overeenkomen met de content binnen het element.

- Ik kan semantische code schrijven, waardoor niet alleen een computer, maar ook mensen de code beter kunnen lezen.

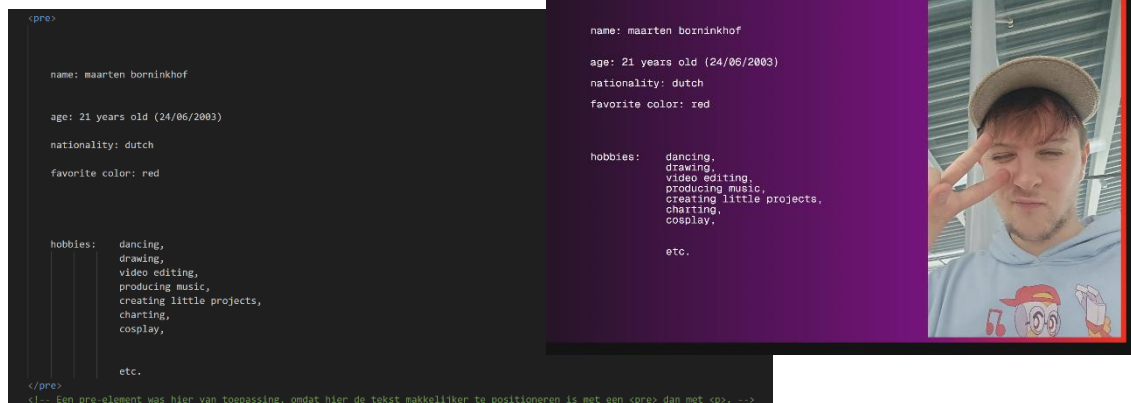
```
<section id="works"> ...
</section>
```

```
<nav class="siteNav"> ...
</nav>
```

```
<header class="header" id="top"> ...
</header>
```

- Ik kan de tags <h1, h6> voor kopteksten, <p> voor paragrafen en <pre> voor vooropgemaakte tekst toevoegen om tekst te weergeven op de website, en weet wanneer ik welke het beste in kan zetten.

Voorbeeld van een <pre>-element.



Voorbeeld van een <h1> en <p>-element onder elkaar.

```
<h1>maarten<br>borninkhof</h1>
<p>&#10022; dancer, designer, more. &#10022;</p>
```

- Ik kan een navigeerbare website bouwen met het <nav>-element (al is deze op mijn website opzettelijk opgebouwd met een <p>-element, waar je deze vaker met een lijst () opgebouwd ziet.

```
<nav class="siteNav">
  <p id="works_li"><a href="#works">works</a></p>
  <p id="impressions_li"><a href="#impressions">impressions</a></p>
  <p id="about_li"><a href="#about">about</a></p>
  <p id="contact_li"><a href="#contact">contact</a></p>
  <br><br>
  <p id="datapunten_li"><a href="datapunten.html">datapunten</a></p>
  <p id="top_li"><a href="#top">to top</a></p>
</nav>
<!-- De fixed navigatiebalk rechtsbovenin. Via hier kun je makkelijk de hele pagina doorkomen zonder te scrollen. -->
```

Ook zie je hier het <a>-element, waarmee ik een link kan leggen binnenin de pagina, om deze website verder navigeerbaar te maken.

- Ik kan een <div>-element toevoegen om geneste elementen te groeperen en makkelijker te stijlen met CSS.

```
<div id="about-content-full">
  <div id="about-content-textWrap">
```

- Ik kan binnen tekst verschillende opmaak toepassen, zoals voor dikgedrukt, <i> voor italics, of <strike> om een streep door de tekst heen te zetten. In dit voorbeeld maak ik de tekst op met *italics* (*schuingedrukt*).

```
et in <i>contact</i> with me?
```

- Ik kan een klikbare link toevoegen om naar email te verwijzen, en kan hieraan target="_blank" toevoegen om deze actie in een nieuw tabblad te laten openen.

```
<a href="mailto:maartenborninkhof03@gmail.com" target="_blank">personal e-mail &#9993;; maartenborninkhof03@gmail.com</a><br>
<a href="mailto:M.Borninkhof@student.han.nl" target="_blank">school e-mail &#9993;; M.Borninkhof@student.han.nl</a>
<!-- mailto: brengt je naar je eigen mailprovider vanwege de target="_blank". Hierdoor kun je makkelijk mij een e-mail sturen. -->
```

- Ik kan code embedden (insluiten) om content van andere websites gemakkelijk toe te voegen. Dit is mogelijk met het <iframe>-element. In dit voorbeeld voeg ik een aantal YouTube-video's toe.

```
<div class="youtubeLinks">
  <iframe width="560" height="315" src="https://www.youtube.com/embed/svAXkATrVJk?si=oSorWp1z1a4nRUNF" title="YouT
  <iframe width="560" height="315" src="https://www.youtube.com/embed/oIgMnA5RBYU?si=xtCMGb6A1PAKR1hb" title="YouT
  <iframe width="560" height="315" src="https://www.youtube.com/embed/sWwW1PLfSRc?si=9wHpi-TB_EyYK25B" title="YouT
  <iframe width="560" height="315" src="https://www.youtube.com/embed/plAOvnEWCnY?si=KE1sOd4fYKrZ93Mp" title="YouT
  <iframe width="560" height="315" src="https://www.youtube.com/embed/MH5Smus8Wqs?si=kh2cGBDnXdsdyh1" title="YouT
  <iframe width="560" height="315" src="https://www.youtube.com/embed/eXMI0EnvIEk?si=stoG6dKaulHsE4YS" title="YouT
  <iframe width="560" height="315" src="https://www.youtube.com/embed/TGkTFvyECOM?si=Ixug8CTOGYZPVII1" title="YouT
  <iframe width="560" height="315" src="https://www.youtube.com/embed/0NiLXA6EamU?si=oa7tkSEd3ukdkPP-" title="YouT
</div>
<!-- Een lijst met Youtube-video's die via insluiten (embedden) makkelijk aan mijn website toe te voegen zijn. -->
```

CSS:

- Ik kan elementen toevoegen aan het CSS-bestand volgens de volgende opbouw:

```
[element] {  
    [hier binnen kan ik het element stijlen]  
}
```

Bijvoorbeeld:

```
p {  
    color: red;  
}
```

- Ik weet hoe ik class- en id-elementen kan toevoegen aan CSS:

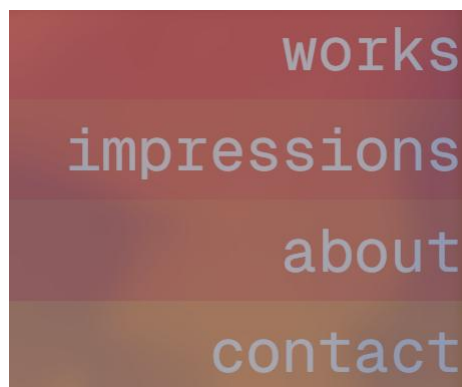
Bij class: . HTML: class="panda" CSS: .panda {}

Bij id: # HTML: id="beer" CSS: #beer {}

- Ik kan veelgebruikte stijlen toevoegen aan een element. Voorbeelden hiervan zijn:

Color	tekstkleur
Background-color	achtergrondkleur
Opacity	zichtbaarheid (0.0=onzichtbaar, 1.0=volledig)
Font-size	grootte van tekst
Position	positie van element ten opzichte van parent
Tekst-align	tekst center, left, right om positie te bepalen
Letter-spacing	de witruimte tussen letters bepalen

```
#works {  
    color: red;  
}  
  
#impressions {  
    color: rgb(255, 55, 0);  
}  
  
#about {  
    color: rgb(255, 81, 0);  
}  
  
#contact {  
    color: rgb(255, 128, 0);  
}
```



- Ik kan aan het html-element 'scroll-behaviour: smooth;' toevoegen, om klikbare navigatie binnen de pagina vloeiend te laten lopen (in tegenstelling tot een instantie transitie).

```
html {
  scroll-behavior: smooth;
  font-family: 'Geist Mono';
}
```

Ook kan ik font-family toepassen om een bepaald font in te stellen. In dit geval betreft het font de hele webpagina, maar zet dit in specifiekere elementen en je kunt meerdere fonts op je webpagina toepassen!

- Ik kan meerdere elementen groeperen om dezelfde stijl toe te passen, door ze te onderscheiden met een komma.

```
#works, #impressions, #about, #contact {
  min-height: 100vh;
  width: 80vw;
  margin: 20px;
  margin-bottom: 200px;
  border: 0px solid green;
  /* Dit gebruik ik om duidelijk in een keer bord
}
```

- In dezelfde wijze kan ik bepaalde elementen binnen een parent-element aanspreken door het achter het parent-element toe te voegen, om verder onderscheid toe te voegen tussen elementen met stijlen.

```
#contact {
  color: orange;
}

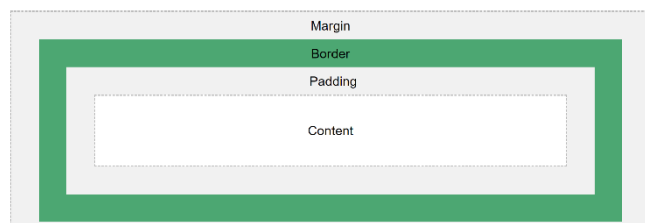
#contact p { Betreft enkel <p> binnenin #contact
  margin-bottom: 20px;
  margin-top: 50px;
  color: grey;
  font-size: 25px;
}
```

- Ik kan :hover toevoegen achter een class of id, om iets te laten gebeuren als je met de muis eroverheen beweegt. In dit geval verandert de kleur en de spacing van de letters.

```
.logo:hover {
  color: ■ rgb(238, 188, 23);
  transition: 1s;
  letter-spacing: 10px;
}
```

Ook kan ik hieraan transition: toevoegen, wat de tijdsduur bepaald van een bepaalde interactie. Zo bewegen de elementen op de site vloeiender.

- Ik kan een box model toepassen met CSS. Hier bepaal ik bijvoorbeeld de padding, margin en border van een element om witruimte of randen toe te voegen om bepaalde content heen.

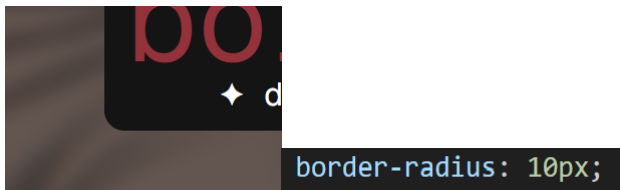


Bron: w3schools

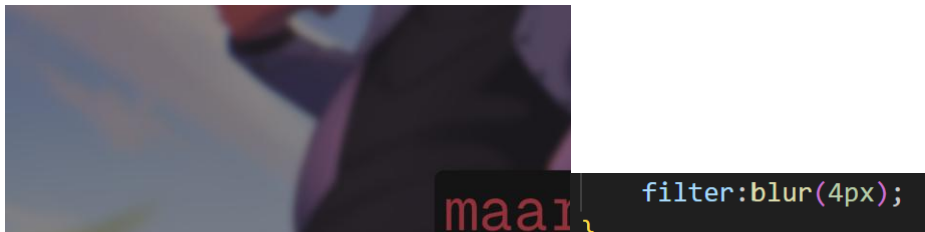
```
#works, #impressions, #about, #contact {
  min-height: 100vh;
  width: 80vw;
  margin: 20px;
  margin-bottom: 200px;
  border: 3px solid ■green;
  /* Dit gebruik ik om duidelijk in een keer borders van meerdere elementen te weergeven. */
}
```



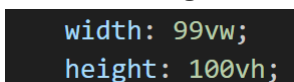
- Ik kan border-radius toepassen om randen van boxen rond te maken.



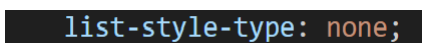
- Ik kan filters toepassen op content. Ik dit voorbeeld voeg ik een blur toe aan de achtergrond.



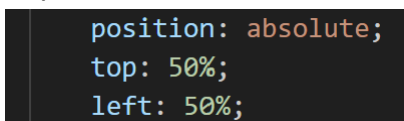
- Ik kan content schalen op basis van breedte en hoogte, en hierbij ook schalen op basis van de grootte van het scherm in plaats van een aantal pixels.



- Ik kan een lijst-type aanpassen, om bijvoorbeeld de puntjes ervoor weg te halen:

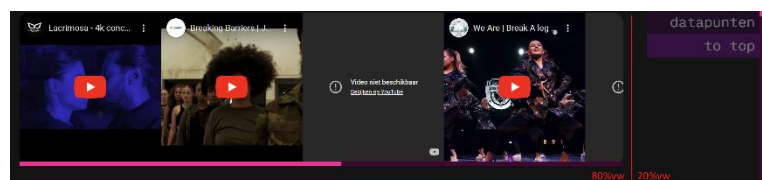
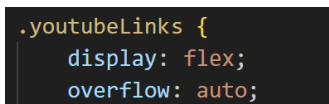


- Ik kan top:, bottom:, right: en left: toevoegen om de positie van content beter te bepalen.



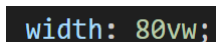
- Ik kan de vorm van display veranderen. In mijn site kun je de volgende terugvinden:

display: flex



(Overigens pas ik hier overflow:

auto toe, waardoor het tot 80vw doorloopt, niet tot rechts van het scherm, maar 80% ervan)

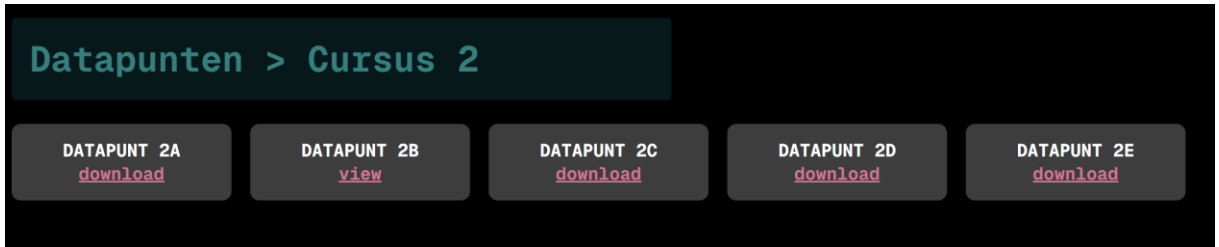


display: inline

```
style="display: inline;">
```

```
/// works ///
```

display: grid



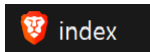
```
#cursus2 .container {  
  display: grid;  
  grid-template-columns:
```

- Als bonus heb ik de scrollbar aangepast, aangezien met de donkere achtergrond deze weg valt.

```
/* Deze volgende vier (4) elementen regelen de opmaak van de scrollbar op de pagina, beide horizontaal en verticaal. */  
  
::-webkit-scrollbar {  
  width: 10px;  
  height: 10px;  
}  
  
::-webkit-scrollbar-track {  
  background: rgb(72, 0, 53);  
  opacity: 0.5;  
}  
  
::-webkit-scrollbar-thumb {  
  background: rgb(255, 0, 153);  
}  
  
::-webkit-scrollbar-thumb:hover {  
  background: rgb(190, 0, 95);  
}
```

Algemene punten voor een goede website:

- Ik kan een goed herkenbaar .html-bestand maken door het 'index.html' te noemen.






- Ik kan een goede map aanmaken met alle benodigde mappen daarin om een webpagina op te bouwen.

📁 _aangeleverd	5-11-2024 15:06	Bestandsmap	
📁 _source	5-11-2024 15:06	Bestandsmap	
📁 css	17-12-2024 03:31	Bestandsmap	
📁 images	18-12-2024 16:14	Bestandsmap	
📁 js	5-11-2024 15:05	Bestandsmap	
📁 pages	5-11-2024 15:05	Bestandsmap	
📁 scorion	8-1-2025 20:46	Bestandsmap	
🛡️ datapunten	10-1-2025 15:19	Brave HTML Docume...	2 kB
🛡️ index	10-1-2025 17:03	Brave HTML Docume...	24 kB

In mijn mappenstructuur staan de .html-bestanden nog niet in de map 'pages'. Dit is op advies van domeindocent Harry, waar hij zei dat je dit pas toe zou moeten passen als de website groter en complexer groeit, of als je bijvoorbeeld verschillende omgevingen heb draaien om je site (zoals een inlogportaal voor managers en werknemers, klanten, etc.).

- Ik kan hyperlinks effectief toepassen en vanuit de teksteditor een map uitkomen (door ../../ toe te voegen), of een bestand in een map juist vinden (/[/mapnaam]/[/bestandnaam.jpg]).
Voor diepgang, zie het toegevoegde mapje 'Oefening HTML Hyperlinks'!
- Ik voeg de hyperlinks op zo'n wijze toe dat je de Showcase Site-map kan verplaatsen en overdragen naar een andere laptop of PC, en dat alle links nog steeds zullen werken. **Ik behandel de Showcase Site-map als de root, niet de C:-schijf van mijn PC.**

- Ik kan een css-reset toepassen, waar ik een bestand toevoeg die een groot deel van de content van een site 'reset' om betere compatibiliteit te bieden tussen verschillende browsers.

 css-reset	17-12-2024 03:31	CSS Source File	2 kB
 datapunten	10-1-2025 15:18	CSS Source File	4 kB
 style	10-1-2025 18:06	CSS Source File	5 kB

Let op!

Er staan hier meerdere CSS bestanden, style.css voor de index.html, en datapunten.css voor datapunten.html. De reden dat ik dit heb gedaan, is omdat ik voor de .html-pagina met de datapunten een andere vormgeving toe wilde passen. De bestanden zijn NIET hetzelfde! Dit geef ik nu aan om te laten weten dat ik begrijp dat het het makkelijkste werkt om één css-bestand te hebben voor je hele website, maar hier geef ik aan dat dit dus opzettelijk is gedaan.

Al deze punten gecombineerd geven in mijn ogen een goed beeld van wat ik allemaal in petto heb. Ik geef aan dat ik weet waar je moet beginnen omtrent schrijven van html- en CSS-code, hoe de syntaxen werken en welke regels en limitaties ze omvatten (html voor de inhoud van een pagina, CSS voor de vormgeving ervan), ik ga er wat dieper op in en pas dingen toe die al wat verder van de basis afliggen, en het belangrijkste, als je op mijn pagina komt vindt je een pagina die **navigeerbaar en interactief is**.

BC2.3.2 Je formuleert heldere leerdoelen die je tijdens de volgende cursussen in de praktijk kunt brengen. [Ontwikkelen]

Met terugblik op cursus 2 neem ik de volgende leerdoelen mee naar de volgende cursussen, beginnend met cursus 3:

Mijn leerdoelen voor cursus 3:

- *Betere tijdsplanning*
- *Vaker en gerichtere feedbackvragen stellen*

Wat ga ik doen?

Ik weet mijn tijd beter in te plannen en weet mij hieraan te kunnen houden.

Hoe ga ik dat doen?

Overzichtelijk maken wát ik moet doen, dit indelen in kleinere blokken om aan te werken, mijn voortgang bijhouden. Ik let op de voortgang en of ik voor, achter of op schema loop. In geval dat ik achterloop pas ik mijn schema aan, met extra tijdblokken om te werken. Ik werk volgens de Pomodoro-methode om lekker productief te kunnen blijven werken.

Wanneer ga ik dat doen?

Gedurende cursus 3.

Wekelijks checken of ik schema loop, en anders werkblokken toevoegen om het op tijd af te krijgen. Ook plan ik zodra ik een nieuwe opdracht krijg tijdig van tevoren deze tijdblokken in, zodat ik goed vooruitzicht kan houden op de rest van de periode.

Wat en/of wie heb ik daarbij nodig?

Notitieblok/checklist of een specifieke app. Doorgaan met het gebruiken van Google Agenda's 'Taken'-functie.

Wat ga ik doen?

Ik stel vaker en beter gerichte feedbackvragen.

Hoe ga ik dat doen?

Met meer frequentie vragen stellen (minimaal 2x per week) bij kleinere aanpassingen en hiermee nagaan wát ik precies wil weten. Ik stel vragen waar ik op kan bouwen. Deze vragen stel ik aan docenten en medestudenten, waardoor ik ook meer transparantie toon in wat ik maak.

Wanneer ga ik dat doen?

Gedurende cursus 3.

Schouw op maandag, en tussendoor bij lessen of zelfstandig werken op de dinsdag en donderdag. De minimale 2 malige keren dat ik deze vragen stel weeg ik af met het progressie van mijn werk, en ik stel deze feedbackvragen als ik zelf moeilijk verder kom, of als ik een aanpassing aan wil brengen waar ik meer dan mijn eigen kijk voor nodig heb.

Wat en/of wie heb ik daarbij nodig?

Medestudenten, domeindocenten, leercoach, document om de vragen met ontvangen feedback in op te slaan.