

```

<div class="homebutton">
  <a href=" ../index.html">Home</a>
</div>

<div class="background">
  <div class="colors">
    <div class="red">
      <p class="textone">Ik ben Joep van Dijk, 20 jaar en ben geboren en opgegroeid in Venray. Hier heb ik het grootste gedeelte van mijn leven doorgebracht en heb ik
    </div>

    <div class="blue">
      <p class="texttwo">Na de Havo heb ik een tussenjaar genomen omdat ik niet wist welke opleiding ik wou doen. Ik heb in deze tijd veel festivals opgebouwd en afgel
    </div>
  </div>

  <div class="colors">
    <div class="green">
      <p class="textthree">Ik hou enorm veel van sporten. Zo heb ik op Judo, Boksen, Rugby en Cheerleaden gezeten. Ik heb met mijn cheerteam op het WK gestaan en we g
    </div>

    <div class="purple">
      <p class="textfour">Ik woon samen met mijn moeder, stiefvader en broertje in Blitterswijck sinds mijn vader is overleden. Hier hebben we een leuke Boerboel (hond
    </div>
  </div>
</div>

```

Dit is het eerste gedeelte van de Html code van mijn selfieweek opdracht. Dit is alleen de tekst die tevoorschijn komt wanneer je op een kleur van mijn gezicht klikt. Ik heb deze classnamen gegeven voor de kleur zodat ik iedere kleur nog kan aanpassen en iedere tekst een classnaam met cijfer gegeven waardoor ik deze later apart kan laten verschijnen met Javascript.

```

43 <div class="portret">
44
45 <?xml version="1.0" encoding="utf-8"?>
46 <!-- Generator: Adobe Illustrator 22.0.1, SVG Export Plug-In . SVG Version: 6.00 Build 0) -->
47 <svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
48 viewbox="0 0 841.9 1190.6" style="enable-background:new 0 0 841.9 1190.6;" xml:space="preserve">
49 <style type="text/css">
50 .st0{fill:#847C7;}
51 .st1{fill:#020304;}
52 .st2{fill:#61C1C;}
53 .st3{fill:#E5201E;}
54 .st4{fill:#58225E;}
55 .st5{fill:none;}
56 .st6{fill:#0B7531;}
57 .st7{fill:#364496;}
58 </style>
59
60 <!-- BLUE -->
61 <path class="st2" d="M618.9,406.4c0,0,19,8.4,33,3,35,5c0,0,7,9,7,11.4-6,2c0,0-3,5-52,3-6,59c-2,4-6,8-5,7-48,7-5,4-60,9
62 s4,3-53,1-1,6-72,6s-6,5-36,3-35,2-69,9s-39,5-45,5-99,7-75c-24,6-9,7-60,7-19,8-65,3-19,8c-4,6,0-20,9,0-20,9,0,13,2,3,51-15,7,2,7
63 13,8-4,61-18,4,3,214,6-4,6c0,0-26,5,2,4-33,6,4,6c-7,2,2-74,2,6,5-84,5,1,4c-10,3-5,1-42,2-16,2-52-15,21-2,2,5,11-17,6-1,9
64 14,6,5,71-26,5,5,415,7,2,71-27,9,21,917,9,0,8c0,0-24,4,43,1-28,4,65,3c-4,1,22,2-5,7,62,6,6,2,76,9c1,9,0,8,4,9,3,8,6,7,3
65 c1,1-3,5,10-34,4,21,4-36,6s10,8-4,1,10,8-4,1s20-24,9,24,6-26,8c4,6-1,9,17,6-7,6,34,4-9,2c0,0,22,5-17,9,37,9-19,5
66 c15,4-1,6,32,5,1,9,38,5,7,9c6,6,12,7,20,3,12,7,20,314,1,0,8c0,0,17,6-10,8,26,8-7,3c10,6,1,6,36,3,13,8,43,3,38,7
67 c7,24,9,9,5,34,3,5,4,52,4c0,0,19,28,9,86,7,57,6c0,0,14,4,9,7,42,11,1C581,322,2,601,346,603,7,350
68 C606,4,354,1,626,5,392,8,618,9,406,4z"/>
69 <path class="st2" d="M165,8,457,8c0,0-13,4-141,2,13,4-171,5c18,8-26,4,28,5-42,2,55,2-46,9c26,7-4,7,49,5-5,1,49,5-5,1
70 s20,6,63,5,59,6,92,1c-35,4,6,5-53,8,7,2-73,3,21,3c-19,5,14,1-11,6,36,8-10,5,38,6s12,6,20,6,61,7,21,3s92,8-4,3,92,8-4,3
71 s-23,1,37,2-23,1,42,2c0,0-20,2,4,3-31,1,3,6c0,0-23,5-27,1-31,4-29,2s-88,8-4,3-100,7-1,8s-22,7,9,4-30,7,17,7
72 s-14,1,18,8-14,1,18,8s170,1,460,165,8,457,8z"/>
73 <path class="st2" d="M166,8,495c0,0-11,6,66,8,28,5,88,1c40,1,21,3,121,3-5,4,121,3-5,4s6,1-1,8,9,7,9c2,9-6,1,17-50,2,12,6-72,2
74 c-4,3-22,27,8-21,7-27,8-21,7,22-9-63,2,2,5174,8,482,4,166,8,495z"/>
75 <path class="st2" d="M331,8,597,2c0,0,38,6,7,2,26-62,8c0,0-7,9,18,5-7,2,22,6c0,7,4,1-8,7,21,1-15,9,26,5l331,8,597,2z"/>
76 <path class="st2" d="M416,7,480,9c0,0,4,52,8,36,1,76,1c35,7,23,2,113,4,7,3,118,6,0,8c6,1-7,5,19,7-13,5,23,5-14,1
77 c2,8-0,4,6,3-1,4,8,1-35,7c1,8-34,3-0,5-55,8-2,3-59,8c-1,8-4-15,2-10,3-29,8-10,3-14,6,0-62,1,2,2-77,3,6,5s-57,4,9,7-70,8,20,9
78 C416,7,473,415,7,469,5,416,7,480,9z"/>
79 <path class="st2" d="M338,654,6c0,0-48,7,54,2-51,6,63,5s-4,7,34,7,14,8,49,1s36,1,23,8,33,9,88,1c0,0,52,4-13,7,75,1-15,2
80 s59,6,2,5,77,6,13,4s25,6,17,7,25,6,17,7s15,5-10,5,37,2-32,5c0,0-2,2-79,4-3,6-94,6s11,9-32,1,11,9-32,116,5-14,11-14,8,3,6
81 c0,0-5,4,1,4-7,6-2,9s-10,2-19,5-32,7-35c-22,5-15,5-44,6-30,47,1-30s,2,9,5,4-22,7,6,1c0,0,1,4,30,1,1,37,2s-7,2,28,5-9,4,30
82 s-71,8,7,9-71,8,7,9s-1,7-2-10,1-9,7c-5,1,2,5-12,6-15,5-12,6-20,6s0,2-15,5,1,5-19,1l338,654,6z"/>
83
84
85 <path class="st3" d="M313,7,741,6c0,0-1,6-10,8,15,2-16,2c16,8-5,4,42,8-5,7,42,8-5,7s30,9-2,2,36,8-3c6,0,8,47,4-5,1,54,4-5,4
86 c7,0,3,19,8-1,1,26,5,0c6,8,1,1,20,7-0,2,21,4,1,9c1,9,5,4-4,6,12,7-20,6,21,1c-16,8,4-38,5,22,2-42,25,2c-3,5,3-17,1,9,7-39,3,9,7

```

Hier zie je het eerste gedeelte van mijn svg's. Omdat dit heel veel paths zijn laat ik niet alle kleuren zien. Ik heb de svg's vanuit illustrator geïmporteerd en deze in mijn Html gezet. Dit waren eerst meerdere svg's met daarin verschillende paths. Wanneer je meerdere svg's over elkaar heen plaatst kun je alleen nog maar klikken op de bovenste, omdat de onderliggende lagen eronder liggen. Ik heb hier lang mee zitten puzzelen maar uiteindelijk lukte het toen ik alle paths in dezelfde svg stopte. Ik moest dan wel elke kleur een aparte classnaam geven zodat ik deze nog kon onderscheiden in de Javascript.

```

1 body{
2   background-color: ■rgb(151, 221, 186);
3   margin: 0;
4   padding: 0;
5   font-family: 'Montserrat', Helvetica, sans-serif;
6   color: □rgb(61, 14, 96);
7 }
8
9 /* HOMEBUTTON */
10 .homebutton, .topbutton {
11   background-color: □black;
12   width: 100px;
13   height: 40px;
14   text-align: center;
15   margin: 50px 0px 0px 90%;
16   position: fixed;
17   z-index: 1;
18 }
19 .homebutton a, .topbutton a {
20   color: ■white;
21   text-decoration: none;
22 }
23
24
25

```

Dit is het css gedeelte van mijn selfieweek project. Hier is wat general styling van de body waardoor deze goed aansluit, de goede fonts gebruikt en de juiste kleuren heeft.

Daaronder heb ik de homebutton geplaatst die ik op al mijn portfolio pages heb waardoor je snel terug kunt naar de homepage van mijn portfolio website.

```

26 /* BACKGROUND */
27 div.colors {
28   display: flex;
29 }
30
31 div.background {
32   position: absolute;
33 }
34
35 div.red {
36   background-color: ■rgb(200, 221, 220);
37   height: 50vh;
38   width: 50vw;
39   display: flex;
40   align-items: center;
41   justify-content: center;
42 }
43
44 div.blue {
45   background-color: ■rgb(186, 186, 228);
46   height: 50vh;
47   width: 50vw;
48   display: flex;
49   align-items: center;
50   justify-content: center;
51 }
52
53 div.green {
54   background-color: ■rgb(221, 189, 214);
55   height: 50vh;
56   width: 50vw;
57   display: flex;
58   align-items: center;
59   justify-content: center;
60 }
61
62 div.purple {
63   background-color: ■rgb(230, 215, 182);
64   height: 50vh;
65   width: 50vw;
66   display: flex;
67   align-items: center;
68   justify-content: center;
69 }
70

```

Hier heb ik de achtergrondkleuren gemaakt. Ik heb gebruik gemaakt van 50vw en 50vh. Dit betekend dat die 50% van hoogte en 50% van breedte pakt. Zo heb je op elke schermgrootte de juiste afmetingen. Ik maak in dit project vaker gebruik van vw en vh en merkte dat dit heel handig is om je projecten er goed uit te laten op meerdere schermgroottes.

In de html had ik één grote div voor alle kleuren, daarin zaten 2 divs. Één voor de bovenste kleuren en één voor de onderste kleuren. Zo kwam iedere kleur op de juiste plek te staan.

```

73 h2{
74   position: absolute; /* Hierdoor is de tekst te lezen over alle andere elementen.*/
75   width: 20vw; /*Geeft de beschrijvende tekst een goede breedte op alle schermen waardoor dit niet over de andere tekst heen komt*/
76   text-align: center;
77 }
78
79 div.tip{
80   display: flex;
81   justify-content: center;
82   padding-top: 60vh;
83 }
84 }
85
86 p.textone{
87   width: 25vw;
88   display: none;
89 }
90 p.texttwo{
91   width: 25vw;
92   display: none; /*Reflectie: Ik merk dat het veel handiger is om met vw en vh te werken ipv px. Hierdoor blijft alles er hetzelfde uitzien
93   ook wanneer de formaat van je beeldscherm verandert. */
94 }
95 p.textthree{
96   width: 25vw;
97   display: none; /*Reflectie achteraf, dit is niet efficiënt code schrijven, deze selectors konden allemaal bij elkaar waardoor je maar 1x width en displat hoefte te schrijven
98 }
99 p.textfour{
100   width: 25vw;
101   display: none;
102 }
103

```

Hier gaf ik de tekst die verteld wat je moet doen de juiste styling en positie. Door vw en vh te gebruiken komt deze altijd mooi onder mijn hoofd te staan zonder dat deze over de andere tekst heen komt. Daaronder heb ik de verschillende teksten de goede breedte gegeven en display op none gezet zodat ik deze met Javascript weer tevoorschijn kan laten komen. Achteraf gezien had ik deze code allemaal bij elkaar kunnen zetten voor meer efficiëntie.

```

104
105 .st2:hover {
106   animation: animatie 2s infinite;
107 }
108 .st3:hover {
109   animation: animatie 2s infinite;
110 }
111 .st4:hover { /*Ook hier had ik de code efficiënter kunnen schrijven zodat ik maar 1x de animatie hoefde op te roepen*/
112   animation: animatie 2s infinite;
113 }
114 .st6:hover {
115   animation: animatie 2s infinite;
116 }
117 .st7:hover {
118   animation: animatie 2s infinite;
119 }
120
121
122
123 /* ANIMATIE */
124 @keyframes animatie{
125   0% {
126     opacity: 1;
127   }
128   50% {
129     opacity: 0.5;
130   }
131   100% {
132     opacity: 1;
133   }
134 }
135
136

```

Hier heb ik gebruik gemaakt van mijn animatie poc. Je ziet hier dat ik voor elke kleur een animatie oproep wanneer je eroverheen hovert. Ook hier had ik mijn code efficiënter kunnen schrijven waardoor je maar 1x de animatie hoeft op te roepen.

Daaronder staat de animatie, ik heb ervoor gezorgd dat de opacity van een kleur steeds wat minder word en weer terug wanneer je met je muis hierover hovert. Hierdoor krijg je nog meer het gevoel dat het klikbaar is en zullen mensen eerder snappen dat je op een kleur kan klikken.

```

141 div.portret {
142     display: flex;
143     justify-content: center;
144     padding-top: 15vh;
145
146 }
147
148 svg {
149     height: 60vh;
150     position: absolute;
151 }
152
153
154
155
156 /* Reflectie: Ik probeerde meerdere svg's over elkaar te zetten en deze apart te laten veranderen met een hover of click. Dit kon niet omdat de hele svg
157 een achtergrond krijg en zodra die over de ander heen ligt de onderste niet meer bereikbaar of klikbaar is. We hebben hier veel sites voor gebruikt maar niet veel werkte
158 https://stackoverflow.com/questions/41079839/hovering-over-transparent-svgs
159 https://css-tricks.com/the-many-ways-to-change-an-svg-fill-on-hover-and-when-to-use-them/
160 https://css-tricks.com/almanac/properties/p/pointer-events/
161 https://www.smashingmagazine.com/2018/05/svg-interaction-pointer-events-property/
162 https://stackoverflow.com/questions/30327743/interactive-svg-hover-over-one-part-change-the-color-of-another
163
164 Uiteindelijk bleek dat je de paths een aparte klasse kunt geven en naar deze kunt verwijzen, eerst verwees ik naar de svg zelf ipv de paths in de svg,
165 omdat ik meerdere svg's over elkaar had liggen moest ik dus alle paths uit alle svg's bij elkaar stoppen in één svg en alle paths klasnamen geven.*/

```

Op het einde van mijn css code heb ik nog de juiste formaat gegeven aan het portret (mijn hoofd) en de svg's waaruit deze bestaan. Ook moeten ze absolute zijn zodat de verschillende svg's goed over elkaar komen te liggen en zodat ze op de achtergrond kleur komen. Ik heb hier ook nog wat proces en bronnen vermeld.

```

Portfolio > 3. derde versie > zelfportret > JS scriptjs > ...
1
2 // BLUE
3 | var st2Elements = document.querySelectorAll(".st2"); // Dit haalt de .st2 classes uit de html en zet deze in een variabele genaamd st2Elements
4
5 st2Elements.forEach(function(st2Elem) { //Voor elke st2Elements word een functie uitgevoerd wanneer je met je mouse erover hovert en wanneer deze weer weg is
6     st2Elem.onmouseover = SetColor; // de functie setcolor zie je hieronder en vervangt de kleur met een andere kleur
7     st2Elem.onmouseout = RemoveColor; // de functie removecolor zie je ook hieronder en haalt weer de kleur weg
8 });
9
10 function SetColor() {
11     st2Elements.forEach(function(e){
12         e.style.fill = "red"; // deze functie zorgt ervoor dat de fill van je svg path verandert naar rood
13     });
14 }
15 function RemoveColor() {
16     st2Elements.forEach(function(e){
17         // (reset
18         e.style.fill = ""; // deze functie zorgt ervoor dat de fill weer weg gaat en de svg zijn originele kleur krijgt
19     });
20 } // omdat elke kleur anders word en is moet ik voor elke kleur deze code herhalen
21
22
23
24 // DARK BLUE
25 var st7Elements = document.querySelectorAll(".st7");
26
27 st7Elements.forEach(function(st7Elem) {
28     st7Elem.onmouseover = SetColor7;
29     st7Elem.onmouseout = RemoveColor7;
30 });
31
32 function SetColor7() {
33     st7Elements.forEach(function(e){
34         e.style.fill = "yellow";
35     });
36 }
37 function RemoveColor7() {
38     st7Elements.forEach(function(e){
39         // (reset
40         e.style.fill = "";
41     });
42 }
43 |
44

```

Dit is het eerste gedeelte van mijn Javascript. In het begin haal ik de .st2 classes uit de html en zet deze in een variabele. Hierna zorg ik dat er een functie word uitgevoerd wanneer je met je muis over deze st2 element hovert en wanneer je er met je muis vanaf gaat. Deze functies staan daaronder.

De functie setcolor word actief als je met je muis over een kleur hovert en geeft deze een andere kleur. De fill van de svg paths worden vervangen door rood in dit geval.

De funtie removecolor is actief wanneer je muis van een kleur afgaat en zorgt ervoor dat deze fill weer weggaat en de path zijn originele kleur krijgt. Omdat elke kleur anders is moet ik deze code herhalen voor alle kleuren in mijn portret. Je ziet hier de code voor blauw en donker blauw, de andere kleuren hebben een gelijke code.

```

93 var st2Elements = document.querySelectorAll(".st2");
94 // vind alle st2 elementen
95 var textone = document.querySelector(".textone");
96 // vind .text element
97
98 var clickstate = false;
99 // maakt een variabele voor de clickstate aan zodat hierin geswitched kan worden
100
101 st2Elements.forEach(function(st2El){
102
103     st2El.onclick = ChangeSomething;
104     // zorgt dat de functie ChangeSomething opgeroepen word wanneer er een click op de variabele st2Elements is
105
106 });
107
108 function ChangeSomething() {
109     if(clickstate == true){
110         clickstate = false
111         textone.style.display = "none";
112     }
113     else{
114         clickstate = true
115         textone.style.display = "flex"; // Eerst had ik nog clickstate en changesomething zonder een 2 erachter, dit betekende dat de clickstate bleef veranderen
116 // ook als er ergens anders geklikt werd waardoor je soms 2x moest klikken om een text zichtbaar te krijgen.
117     }
118 }
119
120

```

Dit is het Javascript gedeelte dat ervoor zorgt dat er een tekst verschijnt wanneer je op een kleur klikt. In het begin haal ik alle st2 classes uit de html en zet deze in een variabele. Dit doe ik ook voor de verschillende teksten. Omdat er meerdere teksten en kleuren zijn moet ik ook deze code herhalen voor de andere teksten en kleuren.

Vervolgens maak ik een variabele voor de clickstate om te kijken of er geklikt is op een kleur of niet. Hieronder roep ik de functie ChangeSomething op wanneer er geklikt word op een .st2 element.

De functie changesomething zet de display op flex wanneer er geklikt is op een kleur en zet deze weer op none wanneer je opnieuw klikt op dezelfde kleur. Omdat ik eerst voor alle codes dezelfde functie had veranderde de kleur soms niet goed. De clickstate veranderde dus steeds.

Om dit op te lossen moest ik ook 4 verschillende ChangeSomething functies maken en 4 verschillende buttonstates. Je ziet hieronder hoe de code van een andere kleur eruit ziet. Ik heb dit dus herhaald voor elke kleur.

```

122 var st7Elements = document.querySelectorAll(".st7");
123 // vind alle st7 elementen
124 var texttwo = document.querySelector(".texttwo");
125 // vind .text element
126
127 var clickstate2 = false;
128 // maakt een variabele voor de clickstate aan zodat hierin geswitched kan worden
129
130 st7Elements.forEach(function(st7El){
131
132     st7El.onclick = ChangeSomething2;
133     // zorgt dat de functie ChangeSomething opgeroepen word wanneer er een click op de variabele st7Elements is
134
135 });
136
137 function ChangeSomething2() {
138     if(clickstate2 == true){
139         clickstate2 = false
140         texttwo.style.display = "none";
141     }
142     else{
143         clickstate2 = true
144         texttwo.style.display = "flex"; // Eerst had ik nog clickstate2 en changesomething2 zonder een 2 erachter, dit betekende dat de clickstate2 bleef veranderen
145 // ook als er ergens anders geklikt werd waardoor je soms 2x moest klikken om een text zichtbaar te krijgen.
146     }
147 }
148

```