



**1** Als je de **Beginner** en **Midden HTML & CSS Sushi Kaarten** hebt afgerond, dan weet je al veel over het programmeren met deze twee talen. Deze kaarten leren je hoe je al die onderdelen samenvoegt om een professioneel uitziende website te bouwen, zoals deze: [dojo.soy/a-html-fin](https://dojo.soy/a-html-fin).

**2** Op de volgende kaarten leer je hoe:

- je het raamwerk van de website maakt
- Je een simpel sjabloon maakt
- je een geanimeerd menu voor je site maakt
- je een geweldig uitziende aanhef maakt
- je een overzicht van je codeerprojecten maakt

**3** Om te beginnen heb je een geraamte van je website nodig. Maak een nieuwe **map** met de naam **gevorderden\_website** aan om je website in te zetten.

Maak binnen die map nog een paar **mappen**:

- **css** - waarin je je opmaak stopt. Het is handig om dit goed te organiseren als je er meerdere hebt voor verschillende dingen als thema's, specifieke opmaak voor bepaalde pagina's etc.
- **img** - waarin je alle afbeeldingen voor je website plaatst, zoals een logo, foto's of screenshots van je projecten.



4

Maak een paar bestanden van waaruit je werkt:

In de map **gevorderden\_website** (de **basis map** van je website) maak je lege bestanden aan met de volgende namen:

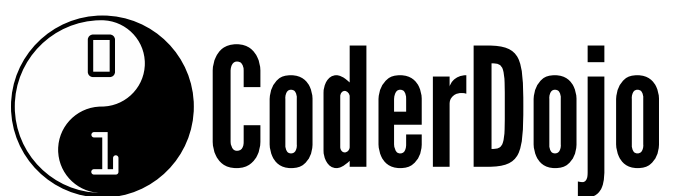
- **template.html** - De opmaakpagina die je naar alle nieuwe pagina's kopieert die je maakt.
- **index.html** - De startpagina van je website.
- **projecten.html** - De pagina waarop je al je projecten opsomt.
- **over.html** - De pagina waarop je informatie over jezelf meldt. In de map **gevorderden\_website/css** maak je een leeg bestand met de naam:
- **style.css** - Het CSS bestand waar je al je stijlen in stopt.

### Kopiëren van sjablonen

Het knippen en plakken van sjablooncodes is niet iets dat professionele webontwikkelaars doen, maar je zult **JavaScript** of een andere taal als **Python** of **Ruby** moeten leren voor je nóg gaver websjablonen kunt maken.

5

Je hebt nu alle mappen gemaakt. In de andere kaarten zul je die mappen gaan vullen met de code waaruit je website bestaat! Een raamwerk als dit maken is een goed begin om met je grote website projecten te beginnen. Het helpt je om alle stukjes te onthouden die later toegevoegd moeten worden wanneer je diep in het programmeren zit.



1

Op deze kaart ga je je sjabloonpagina maken. Als eerste open je **template.html** en maak je de schets van een standaard **html** bestand::

```
<!DOCTYPE html>
<html>
  <head>
  </head>

  <body>
  </body>
</html>
```

2

Nu zet je de **stylesheet** in het **head** gedeelte, en voeg je ruimte toe voor de titel van je pagina. Markeer dit met een **commentaar**.

```
<head>
  <titel><!-- Pagina titel komt hier --></titel>
  <link rel="stylesheet" type="text/css"
href="css/style.css"/>
</head>
```

3

Vervolgens maak je de drie **div** tags die bijna elke webpagina heeft. Geef ze namen die bij hun functie passen (bijv. **class="header"**):

- Eerst een **header** div: Bevat het menu en de websitetitel
- Dan een **content** div: Waar de meeste inhoud van de pagina (zoals je projectoverzicht) staat.
- Tenslotte een **footer** div: Bevat minder belangrijke of standaard links naar bijvoorbeeld een "contact" pagina.



4

Nu is het tijd voor de basisstijlen van je website: Fonts, tekstgrootte, witruimtes, kleuren etc. en om ze op te slaan in je [style.css](#) bestand.

Je kent al genoeg CSS van de Beginner en Midden kaarten om je eigen stijlen te schrijven, maar de volgende kun je zelf nog aanpassen. Let op: ik probeer [headings](#) en [body text](#) duidelijk verschillend te maken.

```
* {  
    box-sizing: border-box;  
}  
  
body {  
    font-family: "Times New Roman", serif;  
    margin: 10px 0 0 0;  
    padding: 0 10%;  
}  
  
h1, h2, h3, h4, h5, h6 {  
    background-color: #49B749;  
    color: #ffffff;  
    font-family: Helvetica, Arial, sans-serif;  
    padding: 0 0 10px 10px;  
    width: 100%;  
}
```

## CSS Eigenschappen

In deze code heb ik een aantal CSS [eigenschappen](#) gebruikt die je eerder bent tegengekomen. Er zijn er echter tientallen. Je komt ze vanzelf tegen. Je kunt ze bekijken op een website zoals [dojo.soy/cssprops](http://dojo.soy/cssprops)



1

De volgende kaarten leren je hoe je een geweldig menu kunt maken voor je website met behulp van **hyperlinks**. Dit is een veel gaver versie van het menu dat je gemaakt hebt met de Beginner Sushi Kaarten. Open eerst **template.html** en voeg een lijst met links toe aan de **header** div tag, als volgt:

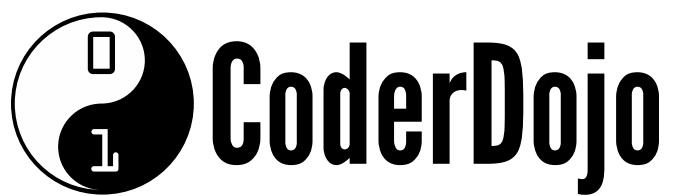
```
<div class="header">
  <nav>
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="over.html">Over Mij</a></li>
      <li><a href="projecten.html">Projecten</a>
    </li>
    </ul>
  </nav>
</div>
```

## Nav Tags

De **nav** tag vertelt de browser (en middelen die slechtzienden gebruiken) dat dit stukje van de pagina de **navigatie** is, welke gebruikt wordt om op de website te navigeren.

2

Laad **template.html** in je browser. Het ziet er niet mooi uit, of wel? Je moet ervoor zorgen dat die lijst er mooi uit komt te zien! Je kunt dit natuurlijk doen door wat **classes** en wat **CSS** toe te voegen. Voeg een **menu** class aan **ul** tag toe en een **menu-knop** class aan elke van de **a** tags.



3

Nu ga je stukje bij beetje de **CSS** bewerken in [style.css](#), zodat je de veranderingen kunt zien.

Zorg er eerst voor dat de lijst zichzelf rangschikt als een menu:

```
.menu {  
  margin: 5px auto 10px auto;  
  list-style:none;  
  text-align: center;  
  display: flex;  
  justify-content: space-around;  
}  
  
.menu > li {  
  display: inline-block;  
}
```

## Flex

Het **display: flex** in **.menu** is erg krachtig. Hoe **flex** precies werkt voor de opmaak van pagina's zou een hele Sushi Serie in beslag nemen, maar je kunt meer hierover zien op [dojo.soy/a-html-flex](https://dojo.soy/a-html-flex).

4

Je hebt een menu dat verbindingen legt tussen je pagina's en het staat op de juiste plek, je kunt het nog gaver maken! De classes staan al op de juiste plek, nu hoef je alleen nog meer **CSS** toe te voegen. Hoe, dat staat op de volgende kaart!



1

Open [style.css](#) en voeg het volgende toe (of draai het om, je weet inmiddels waar je mee bezig bent!):

```
.menu-button {  
  background-color: #0093D5;  
  border: 1px solid #CCCCCC;  
  border-radius: 5px;  
  color: #ffffff;  
  display: inline-block;  
  font-family: "Helvetica Neue", Helvetica, Arial,  
  sans-serif;  
  margin: 5px;  
  padding: 5px 10px;  
  text-align: center;  
  text-decoration: none;  
}
```

### Font Keuze

Merk op dat hier dezelfde lettertypes (fonts) gebruikt worden als bij de koppen op de rest van de website. Dat helpt om de navigatie op te laten vallen.

2

Je gaat een overgang (transition) toevoegen. Ten eerste vertel je **.menu-knop** hoe ermee om te gaan, dus voeg de volgende regel toe aan de **.menu-knop** class in je [style.css](#):

```
transition: all 0.2s ease-out;
```

Dit houdt in dat het 0.2 seconden duurt en langzamer gaat aan het eind.



3

Nu moet je de **:hover** keuzeknop laten veranderen hoe de knop eruit komt te zien als de gebruiker er met de muis over heen gaat:

```
.menu-button:hover {  
    box-shadow: 0 2px 2px rgba(0,0,0,0.2);  
    transform: translateY(-2px);  
}
```

Dit geeft de knop een schaduw en beweegt 2 pixels over de **Y-as** (op/neer).

### Speel er mee!

Verander het gedrag van **hover**. Speel met de grootte, kleur en tijd!

4

Nu heb je een menu! Voeg een titel (**h1**) en logo (**img**) toe, als je dat wilt, aan je **header**, bovenaan het menu. Voeg vervolgens "Gemaakt door [jouw naam]" toe aan de **footer**. Het sjabloon is af! Hoog tijd om de pagina's te gaan bouwen!

5

Sla **template.html** op en kopieer de inhoud. Plak dat in **index.html**, **over.html** and **projecten.html** en sla ze allemaal op.

6

Ga naar **index.html** en voeg een paar paragrafen toe waarin je bezoekers van je website welkom heet. Misschien kun je ze vertellen waar ze zelf kunnen leren om websites te bouwen! Nu heb je een website!





- 1 Nu ga je de [over.html](#) pagina aanpassen door een afbeelding van jouw online avatar toe te voegen (als je er geen hebt, zoek iets gaafs op het internet - ik ben zelf dol op kattenafbeeldingen).

Vergewis je ervan dat de afbeelding vierkant is zodat je er een cirkel van kunt maken met behulp van HTML en CSS! Sla de afbeelding op in [img](#) als [profile\\_pic](#) met de juiste [extensie](#) (afhankelijk van het bestand: .jpg, .png, etc.).

- 2 Op de [over.html](#) pagina, voeg je deze stukjes HTML toe aan je [content](#) div:

```
<div class="picture-heading">
  <h2>Wie ben ik?</h2>
  
</div>
```

- 3 Voeg het volgende aan [style.css](#) toe om de basis van je gave nieuwe kop te maken.

```
.picture-heading {
  height: 150px;
  margin: 10px 0;
  position: relative;
}
```

Hier zet je [position: relative](#) zodat je de [img](#) en [h2](#) tags [position: absolute](#) kan geven, zodat je de één op de ander kunt zetten met de juiste CSS!

4

Bepaal nu de eigenschappen van de afbeelding in **picture-heading**. Je gebruikt **.picture-heading img** (**img** tags binnenin iets met de **picture-heading** class) als een kenmerk hiervoor:

```
.picture-heading img {  
  height: 150px;  
  border-radius: 50%;  
  z-index: 40;  
  position: absolute;  
  left: 20px;  
  border: 2px solid #ddd;  
}
```

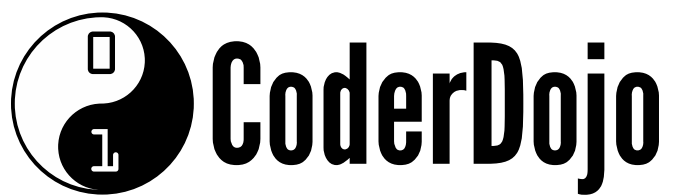
Er zijn dit keer wat interessante stukjes:

- Stel de **border-radius** vast op 50% zodat de afbeelding rond wordt
- De **z-index** bepaalt wat er overheen verschijnt. Hogere komen bovenop.
- Stel de positie in op **absolute** en gebruik dan de **left** eigenschap om het 20 pixels links van de **.picture-heading** te plaatsen.

5

Tenslotte moet je de eigenschappen van de kop zelf instellen.

```
.picture-heading h2 {  
  position: absolute;  
  line-height: 100px;  
  height: 100px;  
  padding: 0 0 0 190px;  
  width: 100%;  
}
```



- 1 Het wordt al wat met je website! Het begint op iets te lijken dat een professionele programmeur zou bouwen! Een laatste truc die je vast al eens op websites hebt gezien. Dit heet de **lightbox**: je klikt op een afbeelding of knop of iets anders, het scherm wordt vager en iets anders (meestal een grotere versie van die afbeelding) verschijnt.
- 2 Je bouwt dit in op je projectenpagina, zodat je dit ongeacht hoeveel projecten je hebt kunt kopiëren. Open [projects.html](#) en voeg aan **content div** het volgende toe (gebruik gerust andere afbeeldingen als je wilt!):

```
<a href="#box1">
  
</a>

<a href="#_" class="lightbox" id="box1">
  <h3>Projectnaam</h3>
  
  <p>Projectbeschrijving</p>
</a>
```

Een **thumbnail** (miniatuur) is de naam van een klein plaatje dat groter wordt als erop geklikt wordt. Je gebruikt er nu één om te leren hoe het werkt, maar je kunt er later meer gebruiken in bijvoorbeeld een **tabel**, om al je projecten te laten zien!



3

En nu de **CSS**. Er zitten een paar speciale stukjes in, die ik later uitleg. Zoals gewoonlijk zet je dit in **style.css**:

```
.lightbox{
    background: rgba(0,0,0,0.8);
    color: #fff;
    height: 100%;
    left: 0;
    position: fixed;
    text-align: center;
    text-decoration: none;
    top: 0;
    visibility: hidden;
    width: 100%;
    z-index: 999;
}

.lightbox:target {
    outline: none;
    visibility: visible;
}
```

- De **lightbox** zit meestal verstopt, gebruik: **visibility: hidden**
- Het heeft **position: fixed**, wat betekent dat het op dezelfde plaats blijft staan, ook al scroll je over de pagina. Met breedte en hoogte neemt het de hele pagina over.
- Zet de **text-decoration** uit om te voorkomen dat alles binnen de **a** tag onderstreept wordt.
- De **lightbox:target** class werkt alleen als de **lightbox** het doel was van de laatste **hyperlink** waarop geklikt is. Dus waar je ook klikt, dit zorgt ervoor dat de **zichtbaarheid** (visibility) weer **verstopt** (hidden) wordt en de **lightbox** ook!