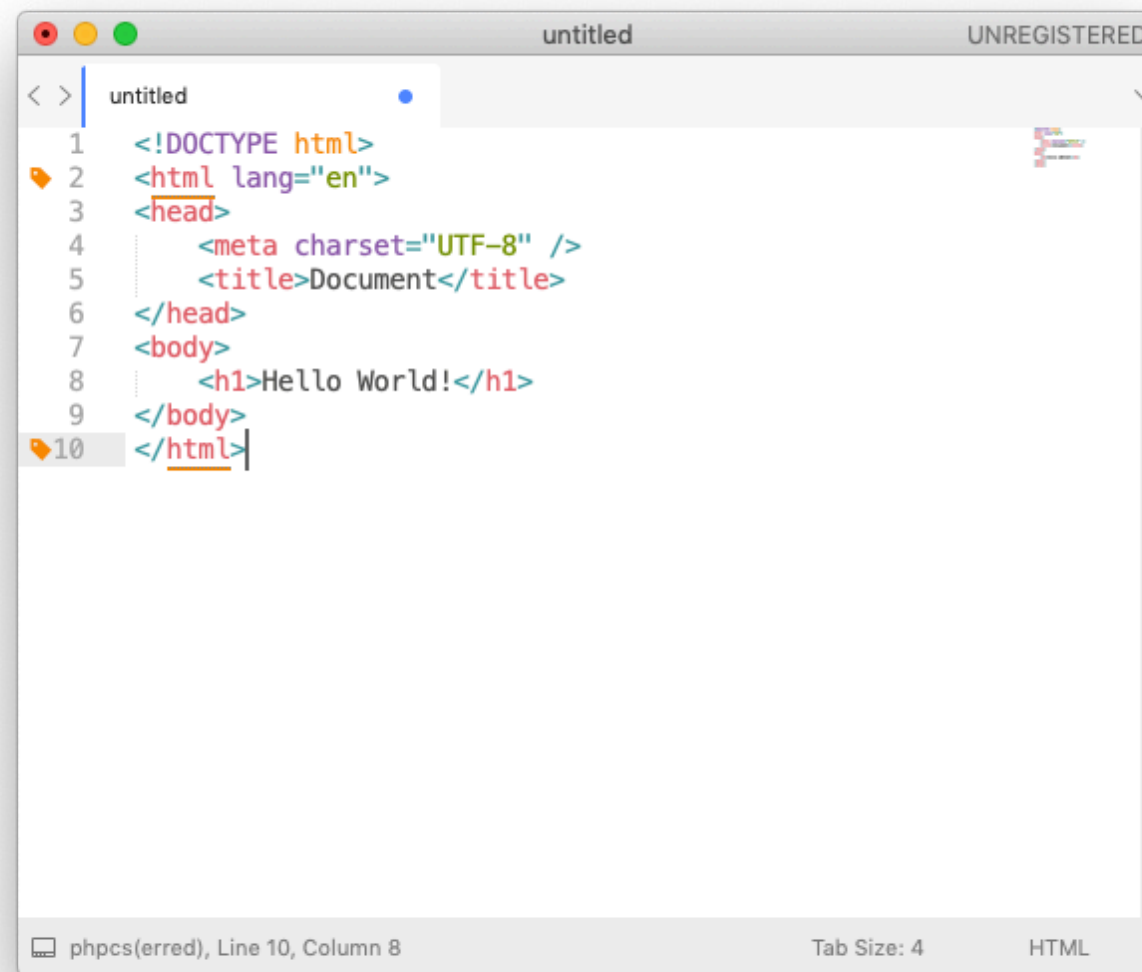


BOOTSTRAP 4 SITE

EEN CSS / JS FRAMEWORK OM SNEL SITE TE BOUWEN



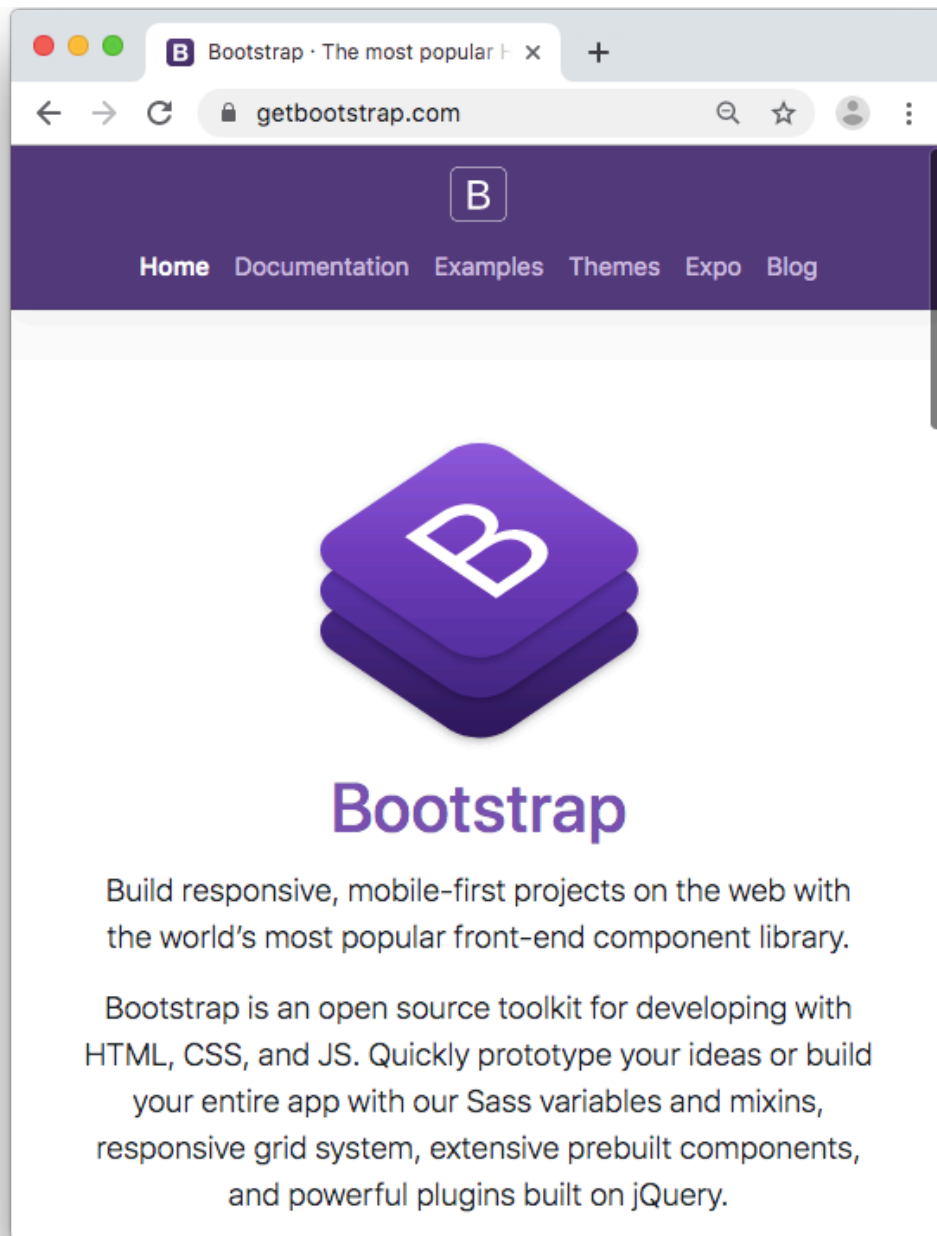
The image shows a screenshot of a code editor window titled "untitled" with "UNREGISTERED" in the top right corner. The editor contains a basic HTML document structure with the following code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <title>Document</title>
6 </head>
7 <body>
8   <h1>Hello World!</h1>
9 </body>
10 </html>
```

The status bar at the bottom indicates "phpcs(errored), Line 10, Column 8", "Tab Size: 4", and "HTML".

PROJECT SETUP

getbootstrap.com



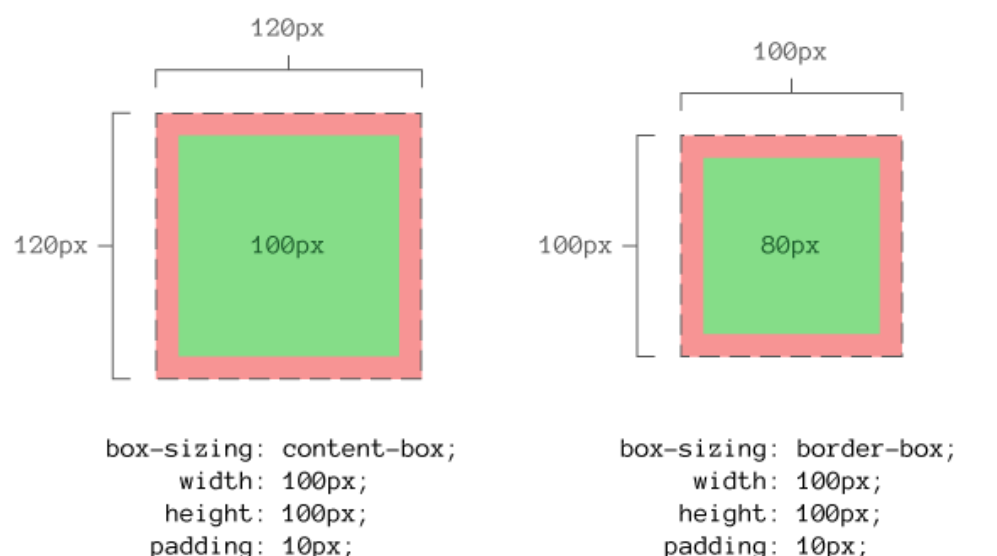
- *Ga naar www.getbootstrap.com*
- *Kies versie 4*
- *documentation*

BOX-SIZING

border-box vs. content-box

CSS Box Sizing

content-box **vs** border-box



● Padding ● Content area ○ Element border

Bootstrap 4 wijzigt de standaard manier van het 'box-model' berekenen naar 'content-box'.
(Waar de padding van de inhoud wordt genomen).

Mocht je problemen krijgen in je CSS daarmee, bijvoorbeeld bij een embedded Google Maps, kan je deze terug zetten naar default:

```
.selector-for-some-widget {  
  box-sizing: content-box;  
}
```

GRID

XX

Wat is het 'Bootstrap Grid'.

Het Bootstrap Grid systeem wordt gebruikt voor layout, specifiek voor Responsive Layouts. (i.t.t 'fixed' 'Utopic Flowers').

Het is cruciaal om te snappen hoe het grid is opgebouwd, namelijk in 'rows' en 'columns'.

De 'row' is de flex-container, de .cols zijn flex-items. Dit is wat we bedoelen met een 'flex'-based grid.

```
<div class="container">
  <div class="row">
    <div class="col">1</div>
  </div>
</div>

<style>
.container {
  width: 100%;
  padding-right: 15px;
  padding-left: 15px;
  margin-right: auto;
  margin-left: auto;
}

.row {
  display: flex;
  flex-wrap: wrap;
  margin-right: -15px;
  margin-left: -15px;
}

.col {
  flex-basis: 0;
  flex-grow: 1;
  max-width: 100%;
}
</style>
```

GRID

de .container

Grid Options

The following table summarizes how the Bootstrap 4 grid system works across different screen sizes:

	Extra small (<576px)	Small (>=576px)	Medium (>=768px)	Large (>=992px)	Extra Large (>=1200px)
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
Grid behaviour	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Container width	None (auto)	540px	720px	960px	1140px
Suitable for	Portrait phones	Landscape phones	Tablets	Laptops	Laptops and Desktops
# of columns	12	12	12	12	12

Het grid-systeem kan zonder de rest van Bootstrap gebruikt worden, als je dat nodig hebt voor je project. Dus zonder de JavaScript of andere CSS onderdelen (components).

Dan moet je enkel het 'bootstrap-grid.css' bestand downloaden en koppelen.

[HTTPS://CODEPEN.IO/PETERVANDENHEUVEL/PEN/LKEPAN](https://codepen.io/petervandenheuvel/pen/lkepan)

GRID

twee kolommen, drie kolommen...



```
<!-- Two columns -->
<div class="container">
  <div class="row">
    <div class="col">Left column</div>
    <div class="col">Right column</div>
  </div>
</div>
```

```
<!-- Three columns -->
<div class="container">
  <div class="row">
    <div class="col">Left column</div>
    <div class="col">Center column</div>
    <div class="col">Right column</div>
  </div>
</div>
```

De Basis van het grid begrijp je nu, je hebt de CSS een aantal slides eerder gezien.

Waarom heb je nu perce die **.container** nodig?

Wat als je de kolommen een andere **breedte** wil geven?

GRID REGELS

the first rules of bootstrap...



```
<div class="row">
  This is very bad, wrong way, no bueno!!
</div>

<div class="row">
  <p>This is also very bad, the wrong way!!</p>
</div>

<div class="row">
  <h2>No headings either! This is the wrong way!!</h2>
</div>

<div class="row">
  <div class="col">Happy :-) This is the right way.</div>
</div>
```

Regel 1:

Kolommen MOETEN een direct kind zijn van een .row

Regel 2:

Row's worden ALLEEN gebruikt om kolommen te bevatten, niets anders.

Regel 3:

Alle Rows worden in een container geplaatst.

De regels zijn belangrijk, de row's en columns werken samen. (Misschien is de negatieve padding al opgevallen).

Gebruik nooit de ene zonder de andere.

GRID REGELS

.container juist gebruiken.

```
<div class="container">
  <div class="row">
    <div class="col">I'm content inside the grid!</div>
  </div>
</div>
```

Container

In het meest basic voorbeeld, dat enkele slides terug stond, zag je dat de .container werd gebruikt om de .row te 'wrappen'.

De .container is de root (het top-level, meest buitenste) element van het Bootstrap Grid.

De Container kan eender welk element of inhoud bevatten. Je bent niet verplicht om het te gebruiken voor .rows of .columns. Bijvoorbeeld je mag ook dit plaatsen:

```
<div class="container">
  <h2>My Heading</h2>
  <div class="row">
    <div class="col">I'm content inside the grid!</div>
  </div>
</div>
```


Op eerste zicht lijkt het erop dat de `.container` niet zo belangrijk is, maar zoals in reguliere website is het erg belangrijk om de breedte van de totale website te sturen. Bovendien stuurt de `.container` ook de negatieve marges die je eerder in de code zag. (op de `.row`)

Bootstrap 4 heeft **standaard 2 soorten containers**:

1) de **'fixed width'** om je inhoud van de pagina te centreren op de pagina. Een beetje zoals je dat vroeger deed:

```
.container{  
    width: 960px;  
    margin: 0 auto;  
}
```

2) een **'full width'**, die over de gehele breedte van de pagina gaat:

```
.container-fluid{  
    width: 100vw;  
}
```

PROJECT SETUP

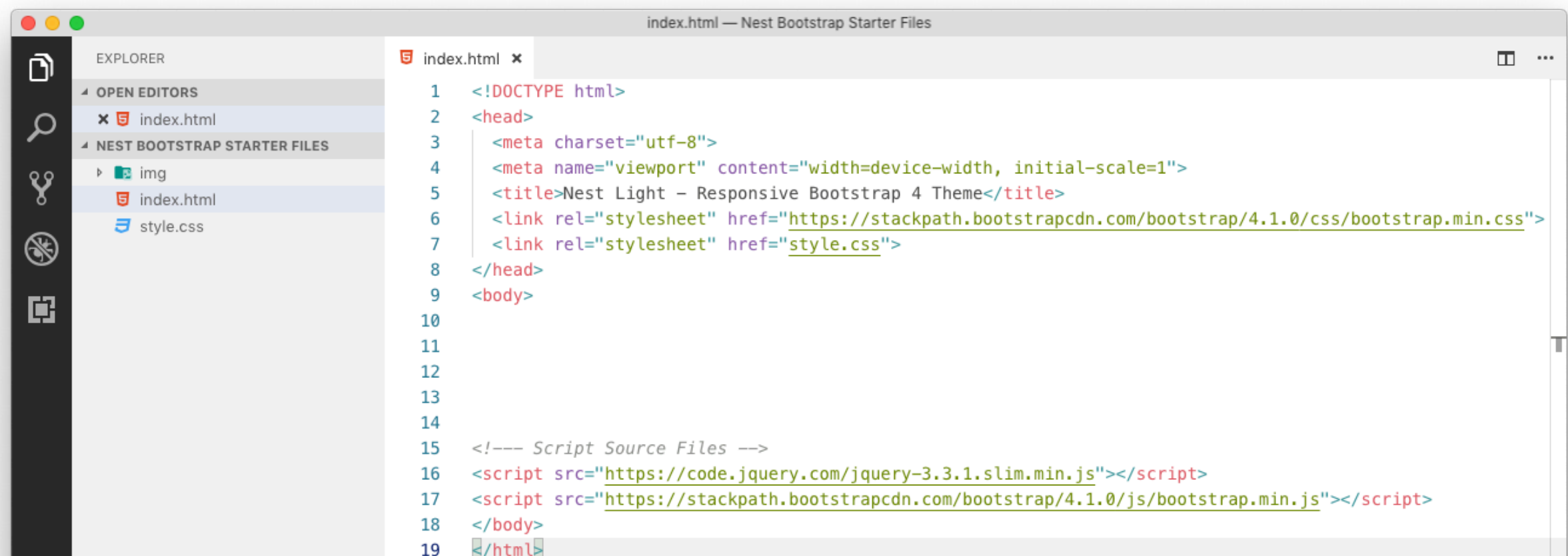
Keuze 1: CDN

Als je een snelle start wil maken, begin je best met het lezen van de introductie pagina.

Dat doe je zo, de <link> tag met de CSS voor al je eigen CSS, en de script-tag NA jQuery en voor je eigen JavaScript.

De meest eenvoudige manier om Bootstrap aan je webpagina's te koppelen is om gebruik te maken van een CDN. Dat wil zeggen dat je de CSS en JavaScript bestanden via een externe link binnenhaalt in je HTML.

In het starter bestand heb ik dit al voor je gedaan.



```
1 <!DOCTYPE html>
2 <head>
3   <meta charset="utf-8">
4   <meta name="viewport" content="width=device-width, initial-scale=1">
5   <title>Nest Light - Responsive Bootstrap 4 Theme</title>
6   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css">
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10
11
12
13
14
15 <!-- Script Source Files -->
16 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
17 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>
18 </body>
19 </html>
```

HTML

smooth scrolling

```
<body data-spy="scroll" data-target="#menu">
```

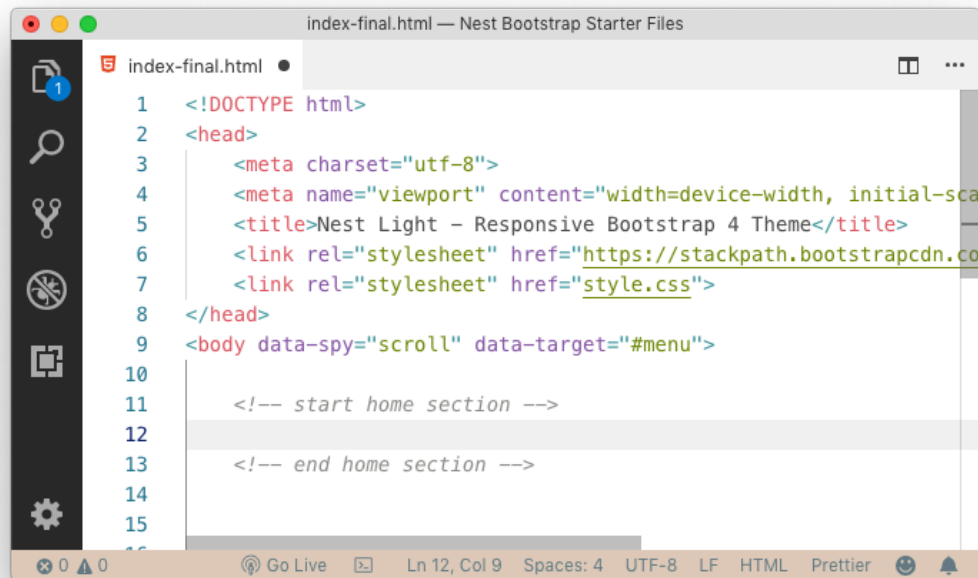
In deze template gaan we de ingebouwde smooth-scroll functie gebruiken. (ScrollSpy).

Dat werkt met links met een ID, en secties met een id van dezelfde naam op een data-attribute

Om te beginnen voegen we een id toe aan de <body> tag, zodat we altijd terug naar boven kunnen scrollen.

HTML

home section



```
1 <!DOCTYPE html>
2 <head>
3   <meta charset="utf-8">
4   <meta name="viewport" content="width=device-width, initial-sca
5   <title>Nest Light - Responsive Bootstrap 4 Theme</title>
6   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.co
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body data-spy="scroll" data-target="#menu">
10
11   <!-- start home section -->
12
13   <!-- end home section -->
14
15
```

We beginnen met het coderen van de “home-section”. Onder de body tag

maak ik commentaar aan:

`<!-- start home section -->`

`<!-- end home section -->`

Vervolgens gebruik ik Emmet om onze HTML te maken:

`div#home`

HTML

navigation

```
<!-- end home section -->

<!-- start nav section -->
  <!-- nav.navbar.navbar-dark.bg-dark.navbar-expand-md.fixed-top -->
  <nav class="navbar navbar-dark bg-dark navbar-expand-md fixed-top">
    test
  </nav>
<!-- end nav section -->
```

Wederom maken we gebruik van Emmet om onze navbar aan te maken:

`nav.navbar.navbar-dark.bg-dark.navbar-expand-md`

Wat doen deze classes:

`.navbar` om onze `<nav>` te stijlen als een navigatie element.

`.navbar-dark` en `.dark-bg` om de achtergrondkleur en de textkleur te sturen.

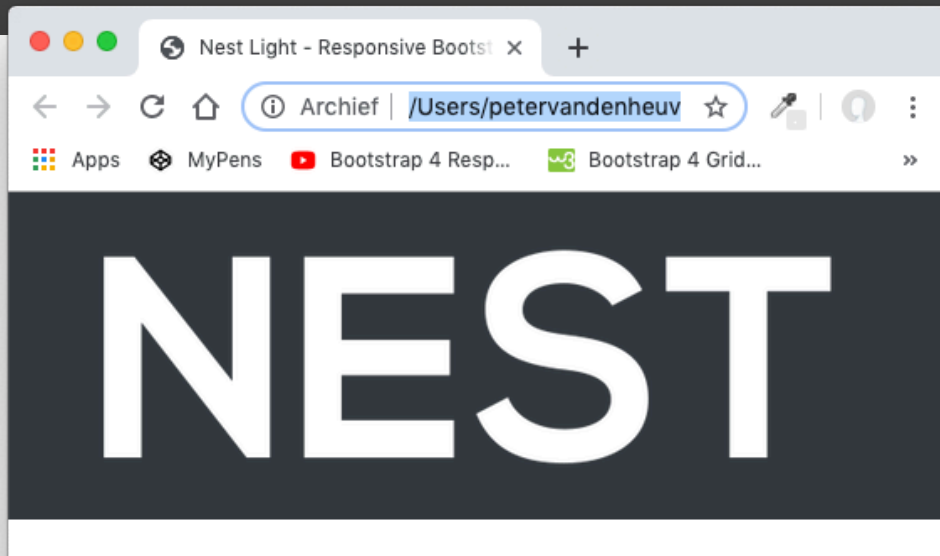
Met `.navbar-expand-md` kunnen we bepalen op welk breakpoint de navigatie volledig zichtbaar is.

`.fixed-top` om hem aan de bovenkant van de pagina vast te pinnen.

Binnen de `<nav>` tags maken we een `.container-fluid` om de elementen over de hele pagina breedte te kunnen uitspreiden.

HTML

logo + navigatie links



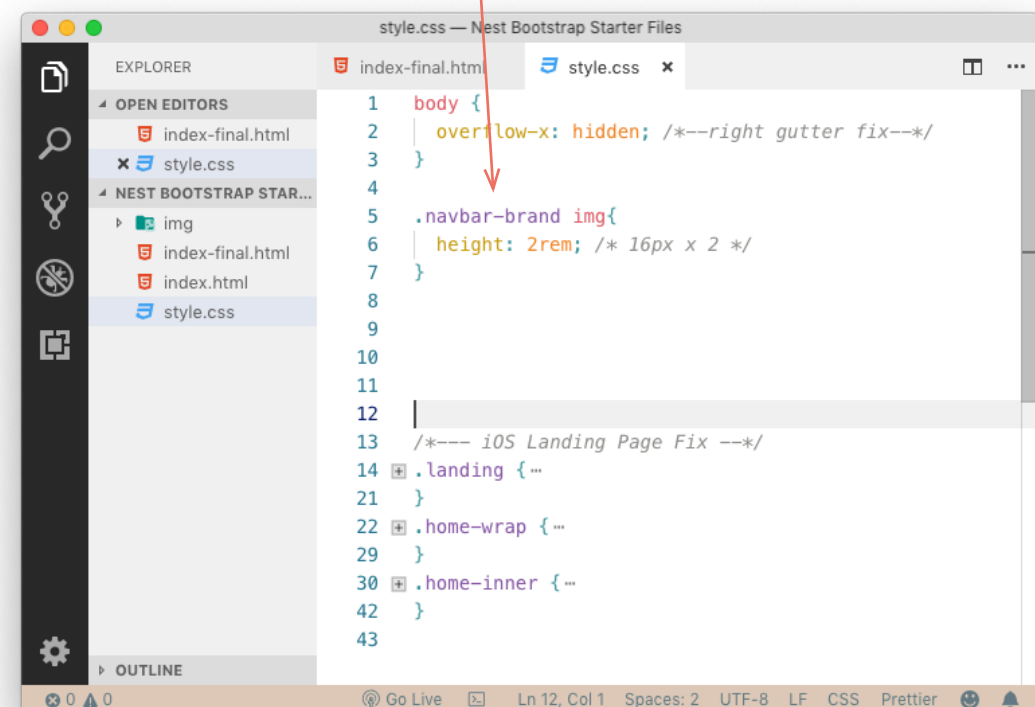
Binnen de `.fluid-container` gaan we verder met het logo en de navigatie links. Dat doen we zo:

`a.navbar-brand > img`

de `src` stellen we in op het `nest.png` bestand.

In `Style.css` passen we de grootte van het logo aan, zodat deze beter meeschaalt.

```
<!-- start nav section -->
<!-- nav.navbar.navbar-dark.bg-dark.navbar-expand-md.fixed-top -->
<nav class="navbar navbar-dark bg-dark navbar-expand-md fixed-top">
  <div class="container-fluid">
    <a href="" class="navbar-brand">
      
    </a>
  </div>
</nav>
<!-- end nav section -->
```



HTML

hamburger menu

We schrijven een button met een aantal attributes achter ons logo:

```
<!-- start nav section -->
<!-- nav.navbar.navbar-dark.bg-dark.navbar-expand-md.fixed-top -->
<nav class="navbar navbar-dark bg-dark navbar-expand-md fixed-top">
  <div class="container-fluid">
    <a href="" class="navbar-brand">
      
    </a>
    <button class="navbar-toggler"
      type="button"
      data-toggle="collapse"
      data-target="#menu">
    </button>
  </div>
</nav>
<!-- end nav section -->
```

button.navbar-toggler

type="button"

data-toggle="collapse"

data-target="#menu"

Om het hamburger icoontje te maken, schrijven we binnen de buttons-tags deze span:

span.navbar-toggler-icon

```
<button class="navbar-toggler"
  type="button"
  data-toggle="collapse"
  data-target="#menu">
  <span class="navbar-toggler-icon"></span>
</button>
```


HTML

het uitklapbare gedeelte

Met emmet schrijven we het uitklapbare gedeelte onder de <button>

`div.collapse.navbar-collapse#menu`

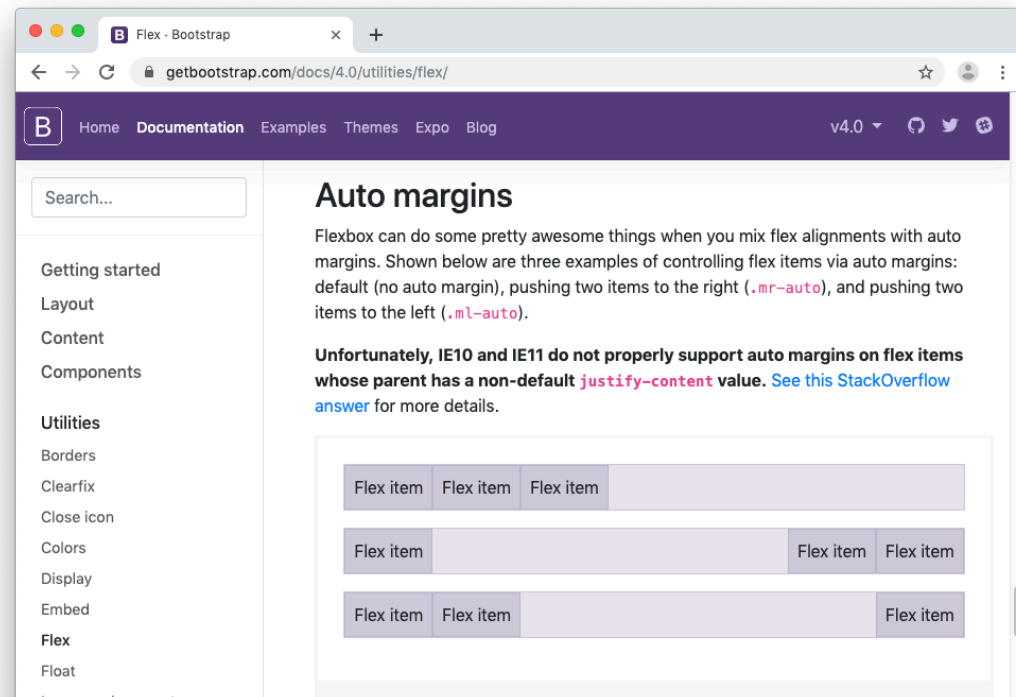
daarop volgend eindelijk onze ul:

`ul.navbar-nav.ml-auto>(li.nav-item>a.nav-link)*5`

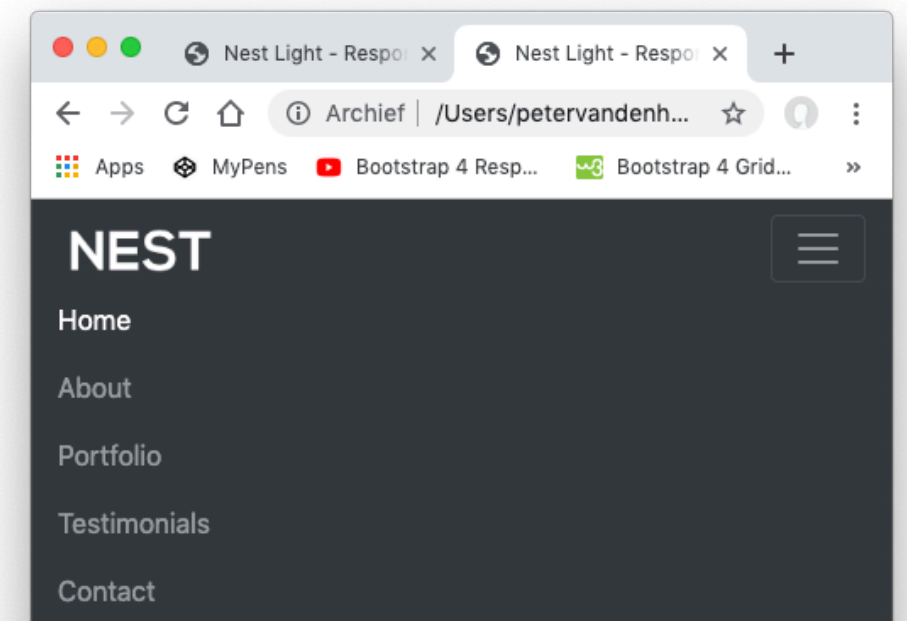
Uiteraard passen we de links en de id's aan.

Waarbij .ml-auto een voor de uitlijning zorgt, meer informatie over deze uitlijnings-classes zie de pagina hiernaast:

<https://getbootstrap.com/docs/4.0/utilities/flex/>



```
</button>
<!-- div.collapse.navbar-collapse#menu -->
<div class="collapse navbar-collapse" id="menu">
  <ul class="navbar-nav ml-auto">
    <li class="nav-item"><a href="#home" class="nav-link">Home</a></li>
    <li class="nav-item"><a href="#about" class="nav-link">About</a></li>
    <li class="nav-item"><a href="#portfolio" class="nav-link">Portfolio</a></li>
    <li class="nav-item"><a href="#testimonials" class="nav-link">Testimonials</a></li>
    <li class="nav-item"><a href="#contact" class="nav-link">Contact</a></li>
  </ul>
</div>
```



HTML

landing -page gedeelte

```
<!-- start section landing page -->
<div class="landing">
  <div class="home-wrap">
    <div class="home-inner"></div>
  </div>
</div>
<!-- end section landing page -->
```

Achter de closing-tag van de `</nav>` beginnen we met de HTML-commentaar tags om duidelijk te maken dat we een nieuwe sectie gaan beginnen.

Daartussen schrijven we deze:

`div.landing>div.home-wrap>div.home-inner`

Er zijn al een aantal voorgeschreven CSS regels in `style.css` die helpen om de layout van dit gedeelte te sturen en te laten werken op alle iOS devices.

We gebruiken straks een CSS-background image om de achtergrond te tonen voor deze `div.landing`.

```
<!-- start section landing page -->
<div class="landing">
  <div class="home-wrap">
    <div class="home-inner"></div>
  </div>
</div>
<div class="caption center-block text-center">
  <h3>Next Responsive Bootstrap Theme</h3>
  <a href="#about" class="btn btn-outline-light">Get Started</a>
</div>
<!-- end section landing page -->
```

HTML

caption + button gedeelte

```
52
53 <!-- end section landing page -->
54
55 <!-- Start Landing Page Caption -->
56 <div class="caption center-block text-center">
57   <h3>Next Responsive Bootstrap Theme</h3>
58   <a class="btn btn-outline-light" href="#about">Get Started</a>
59 </div>
60
61 <!-- About Section -->
```

Achter de 'landing' closing-tag gaan we verder met een nieuw stuk:

Tussen commentaar schrijven we dit:

```
<!-- Start Landing Page Caption -->
```

```
div.caption.center-block.text-center>h3>{Next Responsive Bootstrap Theme}^a.btn.btn-outline-light>{Get Started}
```

```
<!-- end Landing Page Caption -->
```

Op de link voegen we de id toe van '#about', op de href.

HTML

about gedeelte

```
<!-- About Section -->
<div id="about">
  <div class="jumbotron">
    <h3 class="heading">About</h3>
  </div>
</div>
<!-- end of About section -->
```

Wederom een wat langere Emmet-statement:

`div#about>div.jumbotron>h3.heading>{About}`

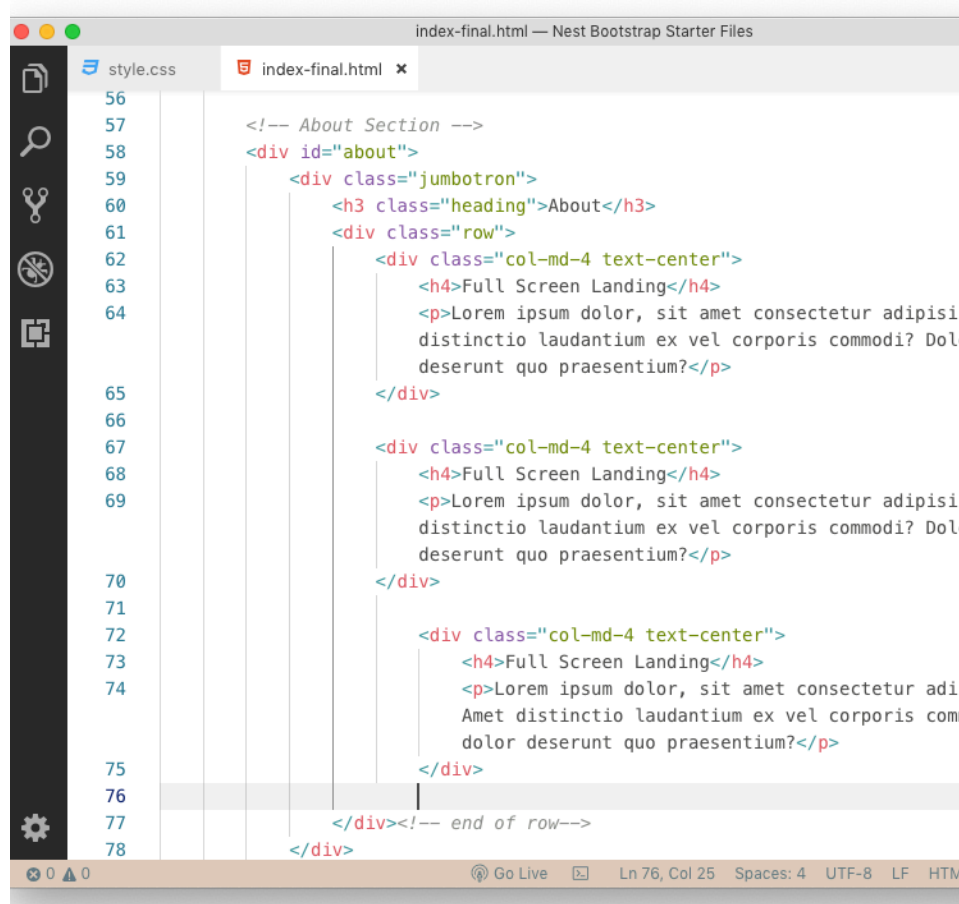
Dat geeft de HTML hiernaast (links).

Achter de closing tag van de `</h3>` maken we een `div.row`

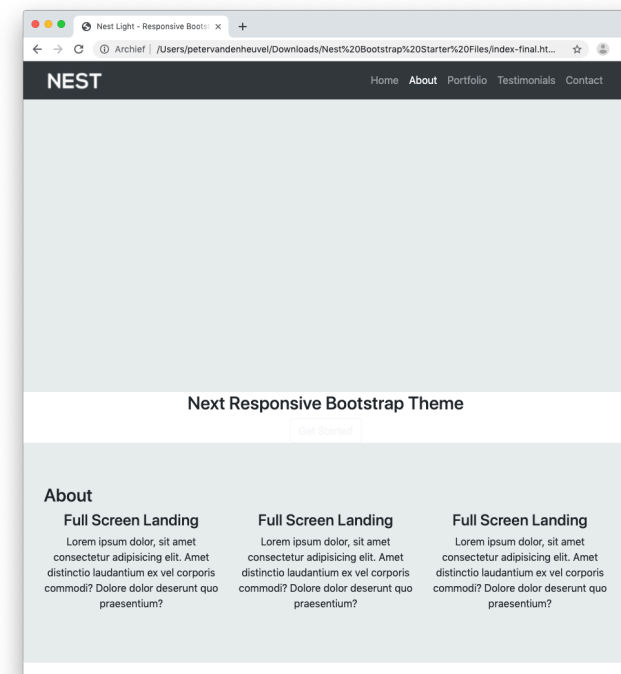
In de `.row` maken we 3 kolommen. Op deze manier:

`div.col-md-4.text-center>h4+p`

Dat doen we 3 keer en we vullen de elementen met text.

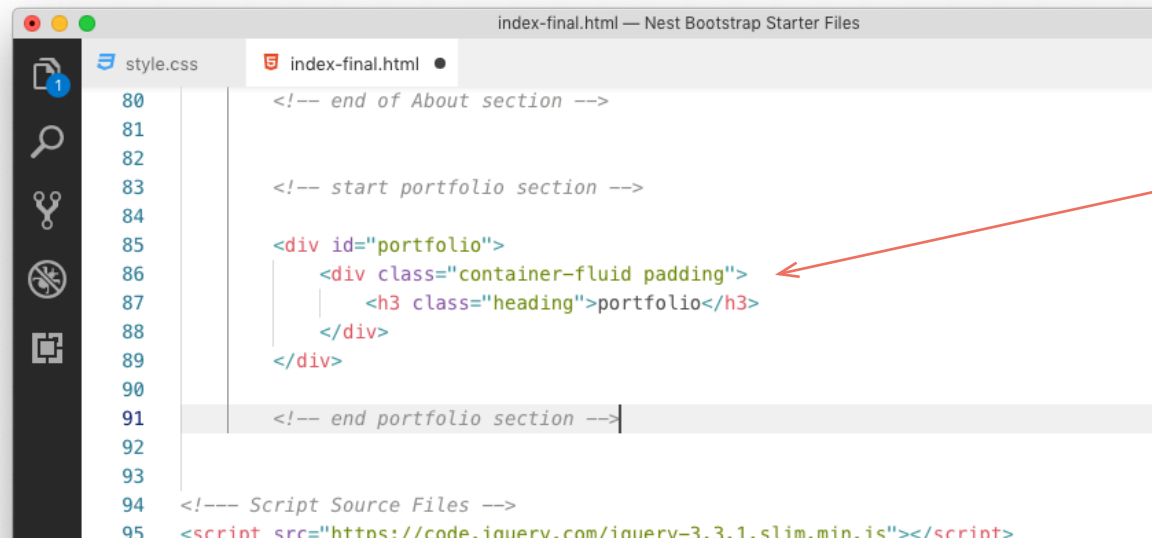


```
56
57 <!-- About Section -->
58 <div id="about">
59   <div class="jumbotron">
60     <h3 class="heading">About</h3>
61   <div class="row">
62     <div class="col-md-4 text-center">
63       <h4>Full Screen Landing</h4>
64       <p>Lorem ipsum dolor, sit amet consectetur adipisci
        distinctio laudantium ex vel corporis commodi? Dol
        deserunt quo praesentium?</p>
65     </div>
66
67     <div class="col-md-4 text-center">
68       <h4>Full Screen Landing</h4>
69       <p>Lorem ipsum dolor, sit amet consectetur adipisci
        distinctio laudantium ex vel corporis commodi? Dol
        deserunt quo praesentium?</p>
70     </div>
71
72     <div class="col-md-4 text-center">
73       <h4>Full Screen Landing</h4>
74       <p>Lorem ipsum dolor, sit amet consectetur adi
        Amet distinctio laudantium ex vel corporis com
        dolor deserunt quo praesentium?</p>
75     </div>
76   </div><!-- end of row-->
77 </div>
78
```



HTML

portfolio gedeelte



```
80 <!-- end of About section -->
81
82
83 <!-- start portfolio section -->
84
85 <div id="portfolio">
86   <div class="container-fluid padding">
87     <h3 class="heading">portfolio</h3>
88   </div>
89 </div>
90
91 <!-- end portfolio section -->
92
93
94 <!-- Script Source Files -->
95 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
```

Achter de closing div van de about sectie, gaan we verder met:

`<!-- start portfolio section -->`

`div#portfolio>div.container-fluid.padding>h3.heading>{portfolio}`

`<!-- end portfolio section -->`

Binnen deze tags schrijven we de kolommen voor de portfolio-items:
(Na de closing tag `</h3>`).

`div.row.no-padding>.col-md-6>img.portfolio`

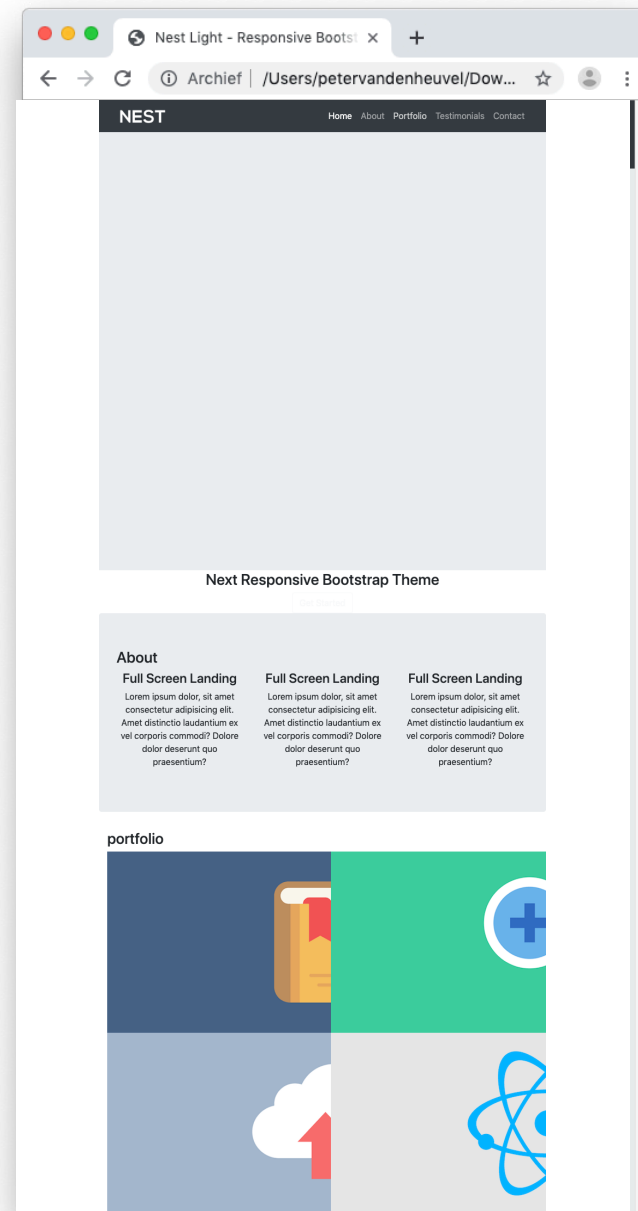
De klassen worden gebruikt voor de styling, en uiteraard passen we de source aan bij de verschillende afbeeldingen.

Vervolgens kopieer ik de `.col-md-6` met inhoud nog 3 maal, zodat we 4 afbeeldingen hebben.

```
<div id="portfolio">
  <div class="container-fluid padding">
    <h3 class="heading">portfolio</h3>
    <div class="row no-padding">
      <div class="col-md-6">
        
      </div>
      <div class="col-md-6">
        
      </div>
      <div class="col-md-6">
        
      </div>
      <div class="col-md-6">
        
      </div>
    </div>
  </div>
</div>
```

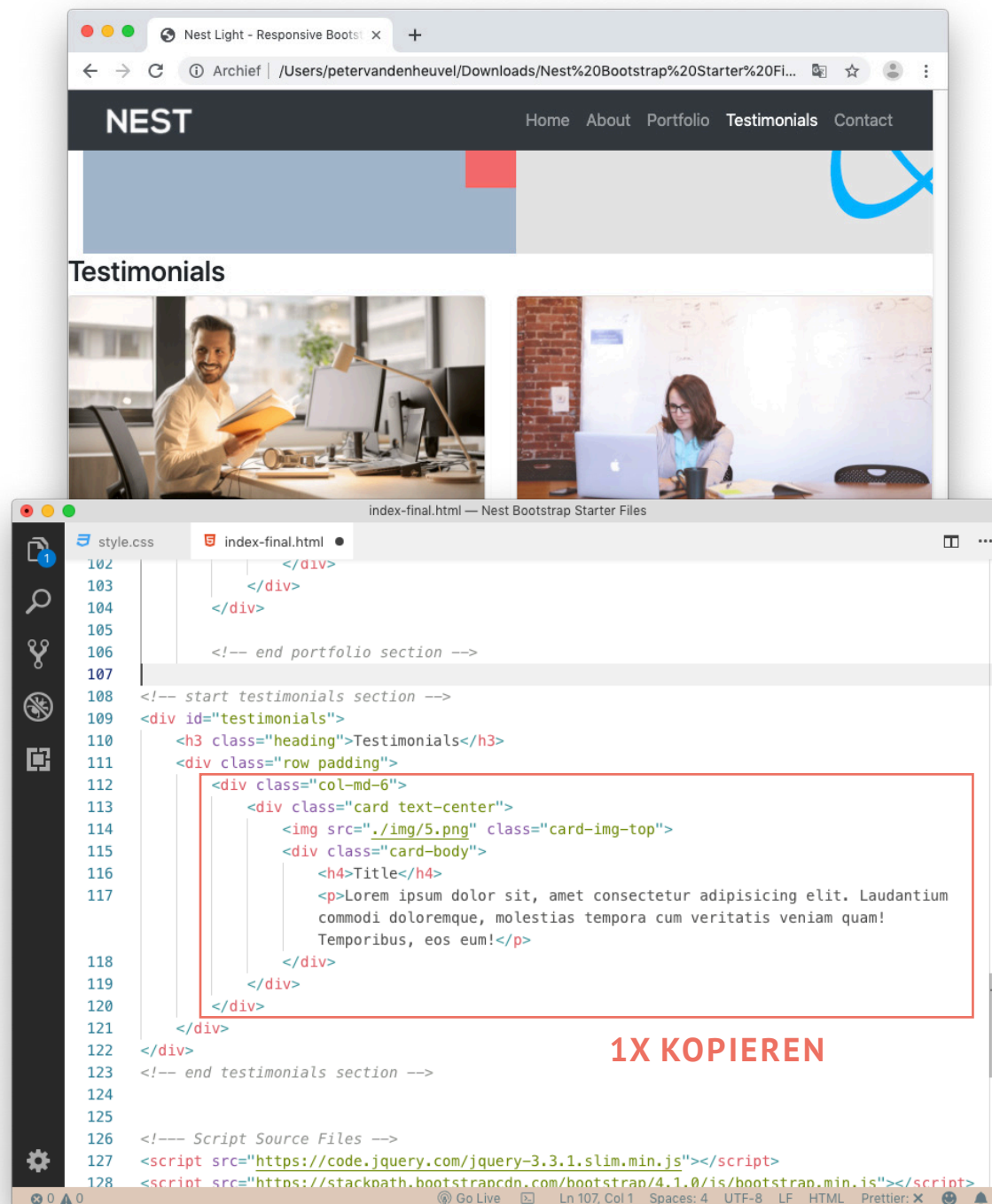
HTML

stand van zaken



HTML

Testimonials gedeelte



You know the drill:

```
<!-- start testimonials section -->
```

```
div#testimonials>h3.heading>{Testimonials}
```

```
<!-- end testimonials section -->
```

Achter de closing tag van de `</h3>`:

```
div.row.padding>div.col-md-6
```

En in de `div.col-md-6`, schrijven we de card op deze manier:

```
div.card.text-center>img.card-img-top+div.card-body>h4+p
```

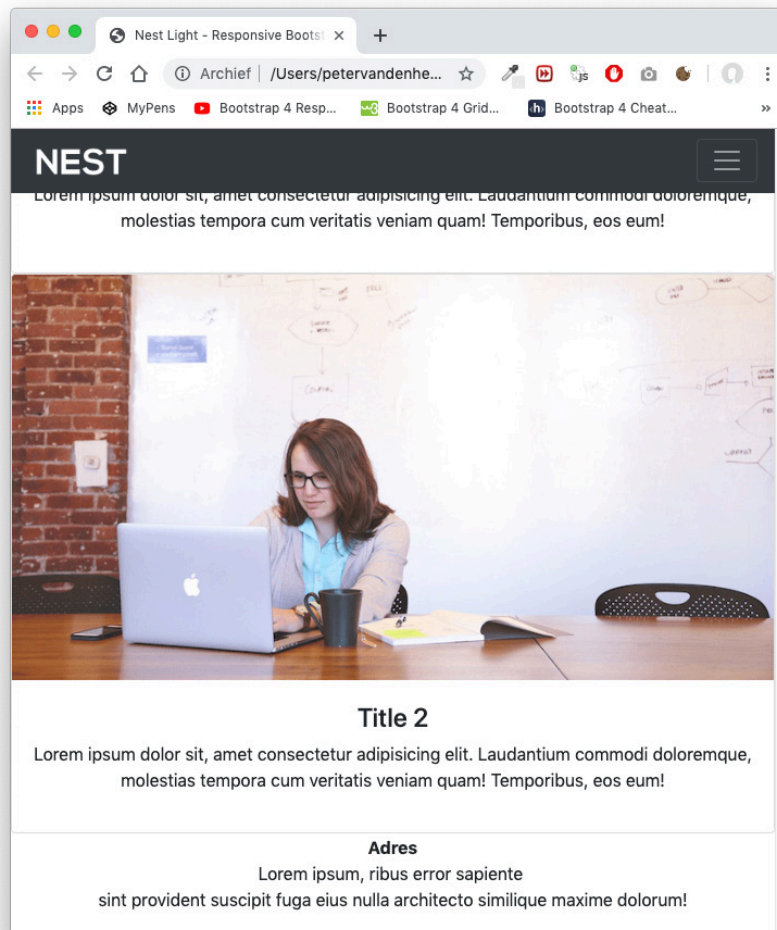
Gebruik 5.png als afbeelding.

Daarna kopiëren we de hele `div.col-md-6` zodat we 2 kolommen met een cardje hebben. Uiteraard pas je ook de afbeelding aan. (6.png).

HTML

contact gedeelte

```
131 </div>
132 <!-- end testimonials section -->
133
134 <!-- start contact section -->
135 <div id="contact">
136   <div class="container-fluid footer">
137     <div class="row">
138       <div class="col-md-3"></div>
139       <div class="col-md-6 text-center">
140     </div>
141   </div>
142 </div>
```



Verder met het laatste gedeelte van de pagina:

```
<!-- start contact section -->
```

```
div#contact>div.container-fluid.footer
```

```
<!-- end contact section -->
```

Binnen de container maken we uiteraard weer kolommen:

```
div.row>div.col-md-3+div.col.md-6.text-center
```

We gebruiken een lege kolom (.col-md-3) om de andere in het midden te plaatsen op grotere schermgroottes.

De inhoud binnen de .md-6:

```
img+p>lorem20
```

pas de lorem tekst aan met een fictief adres, als je dat liever ziet. Je kan gebruik maken van de en
 tag's.

CSS

add some styling to it

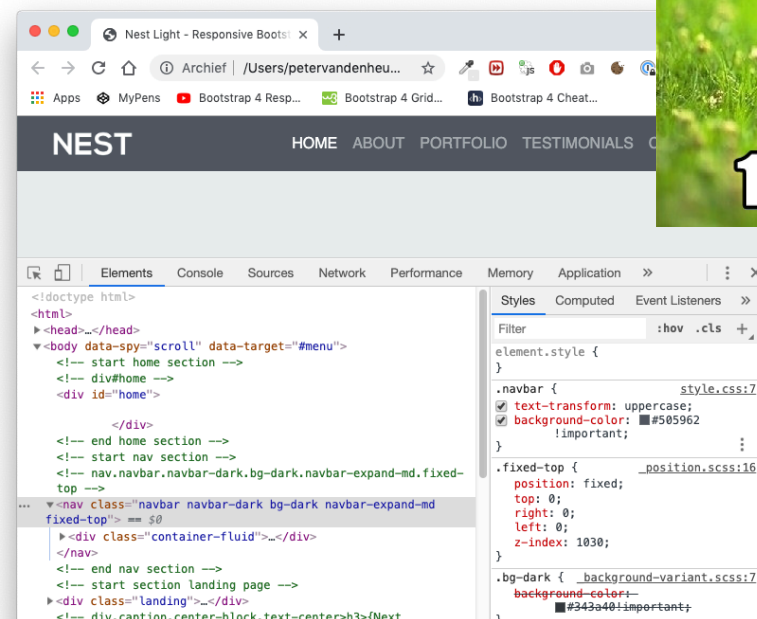
CSS

style your <body>

```
body {  
  overflow-x: hidden; /*--right gutter fix--*/  
  font-family: sans-serif;  
  color: #505962;  
}
```

```
.navbar {  
  text-transform: uppercase;  
  background-color: #505962 !important;  
}
```

```
.navbar-brand img {  
  height: 2rem; /* 16px x 2 */  
}
```



De **body** krijgt een default lettertype. Uiteraard kan je er ook voor kiezen om hier een Google Font te gebruiken.

De **text-color** wordt een wat minder zwaar zwart.

Bij **.navbar** schrijven we een text-transform om alle letter in hoofdletters te plaatsen. De achtergrondkleur is hetzelfde zwart als de vorige CSS regel.

Als je de standaard versie van bootstrap gebruikt zal je in de inspector zien dat de background kleur van de <nav> een !important-tag mee krijgt. In dit geval is dat dus niet te overriden met CSS, behalve met een !important-tag. **Uiteraard is dit af te raden**, en dan is het beter om je Bootstrap opnieuw te compilen met SCSS/SASS, naar je eigen wensen.

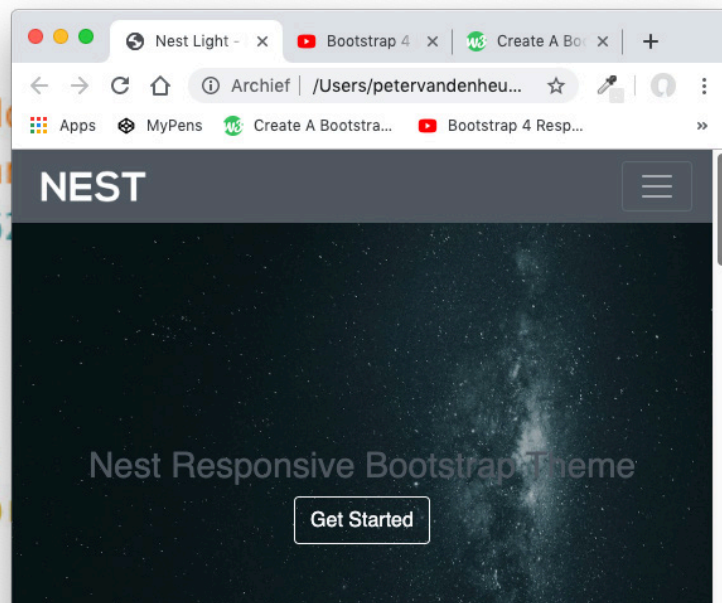
Onthoud, 'If you use !important, some where a kitten dies'...

CSS

style the hero

```
body {  
  overflow-x: hidden;  
  font-family: sans-serif;  
  color: #50596b;  
}
```

```
.navbar{  
  text-transform: uppercase;  
  background-color: #50596b;  
}
```



```
.navbar-brand img{  
  height: 2rem; /* 16px x 2 */  
}
```

```
.home-inner{  
  background: url(../img/landing.png);  
}
```

```
.caption{  
  width: 100%;  
  max-width: 100%;  
  position: absolute;  
  top: 40%;  
  z-index: 1;  
}
```

We gaan nu aan de slag met de **.caption**.

De breedte wordt op 100% gezet, en een max-width van 100%;
Position: absolute om hem te kunnen uitlijnen en bovenop de pagina te plaatsen.

(Hij is uit de flow, en wordt nu uitgelijnd tov de body.)

Met de **z-index** garanderen we dat hij altijd bovenop staat.

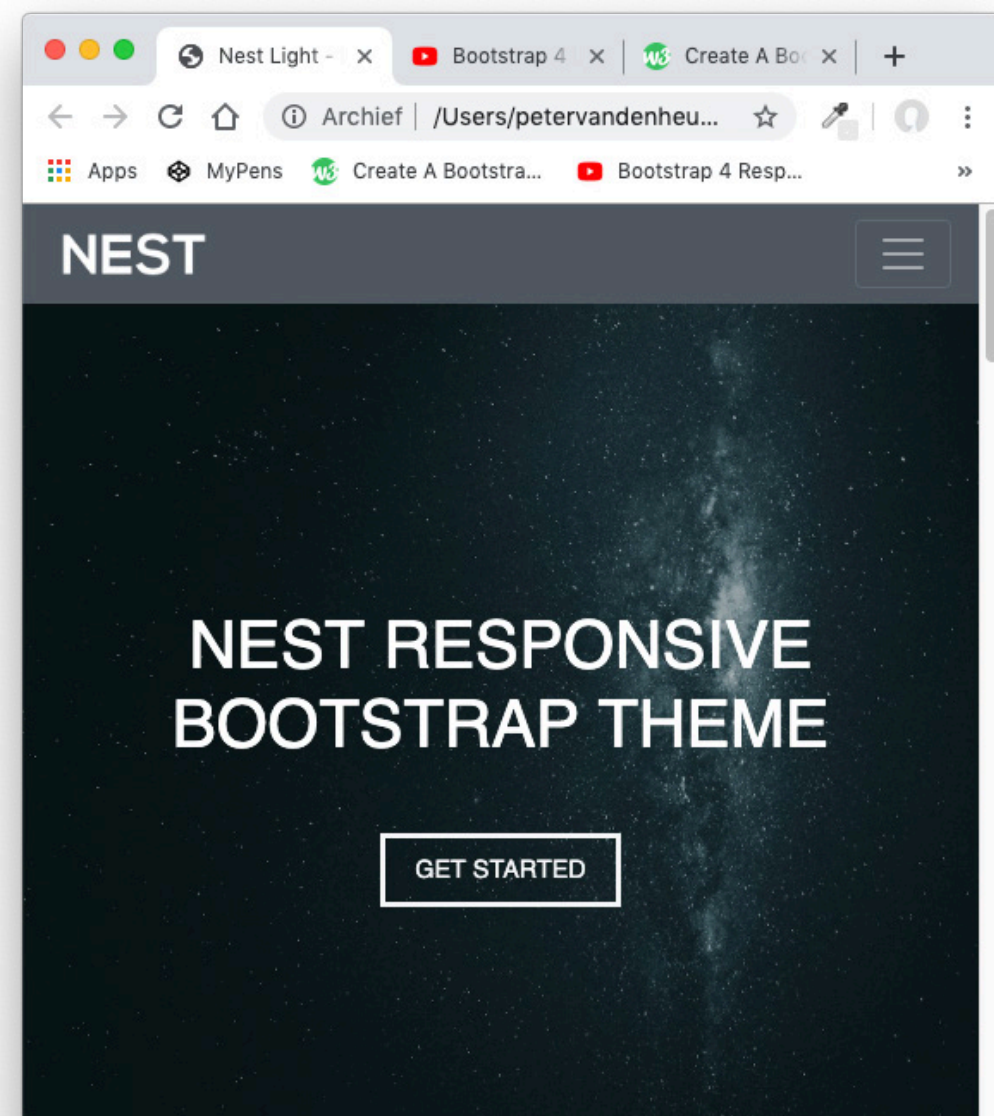
CSS

style the header

```
27 .caption h3{
28   color: #fff;
29   font-size: 2.4rem;
30   text-transform: uppercase;
31   padding-bottom: 2rem;
32 }
```

```
33
34 .btn{
35   border-width: medium;
36   border-radius: 0;
37   text-transform: uppercase;
38   padding: 0.5rem 1.1rem;
39   font-size: 0.9rem
40 }
```

Dit spreekt uiteraard voor zich. De font-size is relatief tov. de root-element. En zal dus niet 'stacken', itt. em's.



CSS

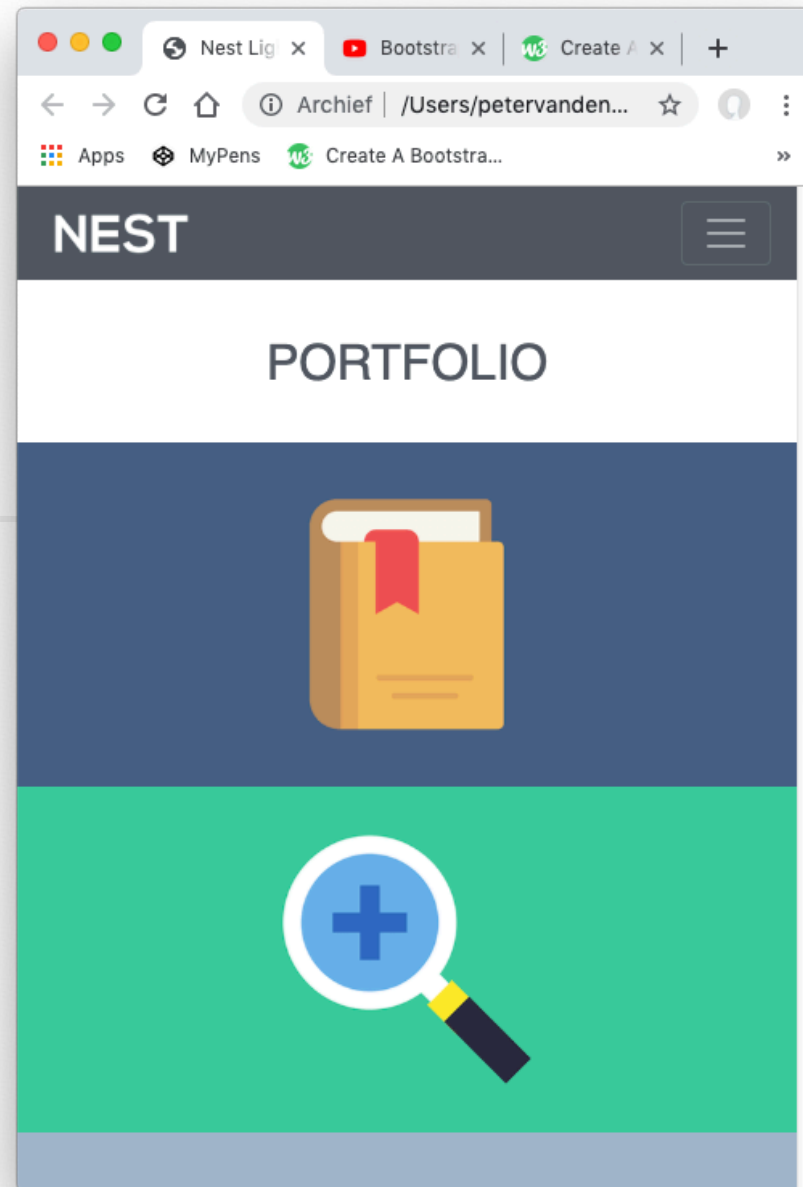
style the heading and helper-class

```
42  h3.heading{
43    text-align: center;
44    font-size: 1.9rem;
45    text-transform: uppercase;
46    margin-bottom: 1.9rem;
47  }
48
49  .padding{
50    padding: 2rem 0;
51  }
```

CSS

style the heading and helper-class

```
52  
53 img.portfolio{  
54   max-width: 100%;  
55 }  
56  
57 .no-padding .col-md-6{  
58   padding: 0;  
59 }
```



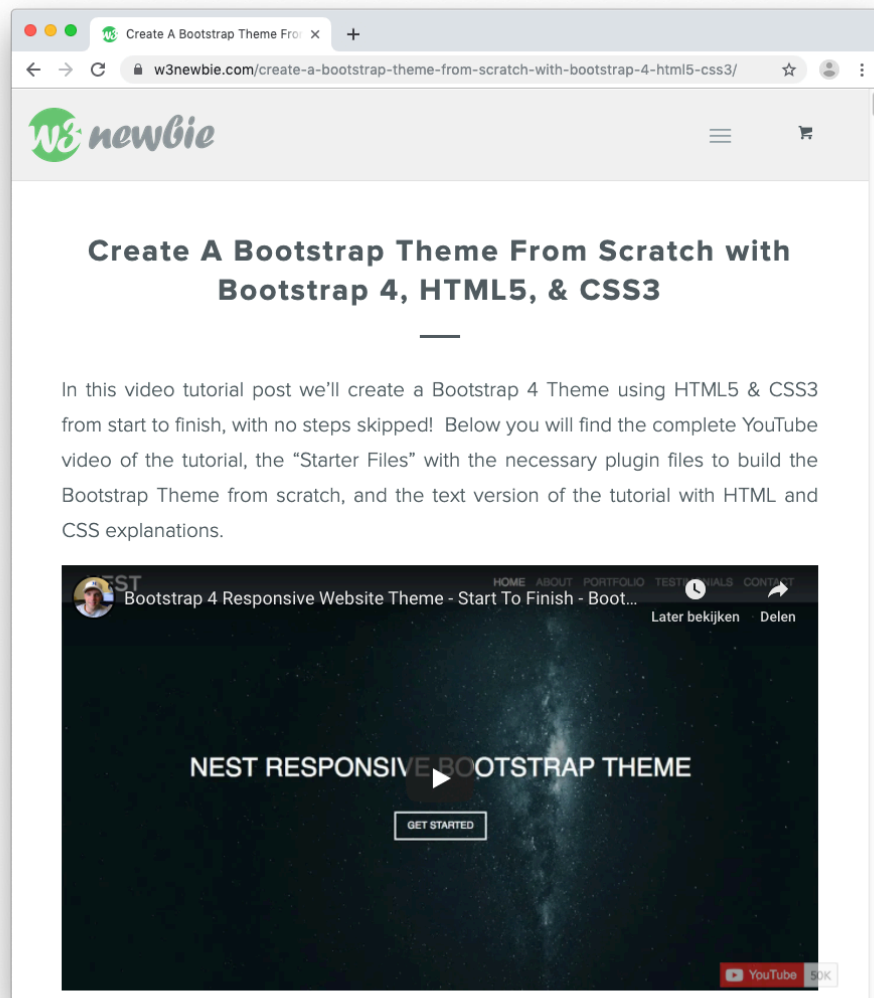
CSS

cards & footer

```
60
61  .card {
62  |    margin: 2rem;
63  |  }
64
64
65  .footer .row {
66  |    background-color: #505962;
67  |    padding: 1rem 2rem 3rem;
68  |    color: white;
69  |  }
70
71  .footer img {
72  |    height: 2rem;
73  |    margin: 1.5rem 0;
74  |  }
```

EINDE

bron vermelding



De vertaling van deze tutorial komt van deze website:

<https://www.w3newbie.com/create-a-bootstrap-theme-from-scratch-with-bootstrap-4-html5-css3/>

Wil je zijn stappen volgen kan je dit YouTube filmpje volgen:

<https://www.youtube.com/watch?v=Ht1F9KuhJO4>