

# Godt i gang med Bootstrap

## Indledning

Bootstrap er et frontend framework, dvs. en samling af HTML, CSS, og JavaScript/jQuery, der kan bruges til at gøre udviklingen af din frontend hurtigere. Bootstrap er specielt godt til responsive websites, dvs. websites der tilpasses bredden af din viewport. Da Bootstrap er et frontend framework, må backend-delen laves med andre løsninger, f.eks. php/mysql, WordPress, Codeigniter mm. Den slipper vi for i Digital Æstetik.

Bootstrap er noget af det mest populære indenfor udvikler-verdenen lige nu. Der er flere galleri-sites, der viser sites lavet med bootstrap, eksempelvis <http://www.lovebootstrap.com/> og <http://builtwithbootstrap.com>

Bootstrap er et responsivt design framework, som har en "mobile-first" tilgang til webudvikling. Det betyder at man først fokuserer på layout på mobilen, og dernæst skalerer sitet op til større skærme.

Bootstrap kan downloades på <http://getbootstrap.com>, hvor der også er mange flere informationer om at komme i gang med bootstrap.

Hvorfor et frontend framework? Det speeder udviklingshastigheden op! Man kan lave færdige sites og prototyper hurtigt (når man først kender systemet)! Til gengæld "bloater" et framework ofte koden, i forhold til hvis man laver alt fra bunden.

## Bootstrap 3

19. August 2013 blev Bootstrap version 3 lanceret. Det har betydet en række mindre ændringer i forhold til version 2. Har du arbejdet med version 2, er der derfor god mening i at hænge på, og prøve version 3 denne gang.

Først og fremmest hopper Bootstrap 3 med på Mobile First bølgen. I version 2 skulle man tilføje responsive features. Med udgave 3, derimod, består Bootstrap af ét responsiv stylesheet, baseret på et "fluid" grid bestående af 12 kolonner. På samme tid har Bootstrap 3 også fået en række moduler, der specielt er rettet mod at lave sektioner formatteret til mobilen.

## Download bootstrap

Download bootstrap ved hjælp af den store "Download Bootstrap"-knap på forsiden af [getbootstrap.com](http://getbootstrap.com). Der er tre mapper i downloadet: css, fonts og js. Læg disse mapper ind i et nyt projekt/nyt webhotel.

OBS! Trykker du på "Download source" knappen, downloader du ud over bootstrap også en masse hjælpedokumenter: Eksempelfiler, read-me filer mm. Det kan være godt at kigge dem igennem, men det kommer vi først til senere. Så lige nu er det nemmere at starte med indholdet fra "Download Bootstrap."

Der er også et andet alternativ: På "Customize" fanen kan man også konfigurere og downloade en tilpasset udgave af bootstrap. Dermed kan filerne gøres mindre. Det vender vi tilbage til senere - lige nu bruger vi den fulde udgave af bootstrap, som downloades på forsiden.

## CSS-mappen

I CSS-mappen er der fire filer:

- bootstrap-theme.css
- bootstrap-theme.min.css
- bootstrap.css
- bootstrap.min.css

Der er egentlig blot tale om to filer - hver af dem i en "normal" udgave og en "minified" udgave. Minified filer (dem der hedder min.css), er filer hvor alle overflødige tegn, linjeskift og mellemrum er fjernet. De fylder mindre, og er dem man inkluderer i sin html-filer. Den normale CSS-udgave kan så bruges hvis man skal checke op på koden i filen, da den er let læsbar, som normal CSS. Theme.css filerne er et flat-design layout tema, der følger med bootstrap. Det anvender vi ikke (til at starte med).

## Fonts-mappen

I fonts-mappen ligger der en font i fire versioner. Der er også mange ikoner i fonten - "glyphicons." Det er en stor fordel at have ikoner som del af en typografi, da de dermed kan skaleres op og ned (font = vektor). Det er en stor fordel, når der skal arbejdes med forskellige skærmstørrelser, skærmopløsninger mm. (f.eks. Retina-skærme med høj opløsning).

## JS-mappen

Indeholder bootstrap.js og bootstrap.min.js, som i bund og grund bare er en masse JavaScript-filer, og kernen i bootstrap. Den minifiede fil inkluderes.

# Bootstrap dit site

For at bootstrappe dit site, er der et par filer, der skal inkluderes - et stylesheet og en JS-fil. Det kan gøres på flere måder, alt afhængig af hvordan du vil arbejde:

A) Downloade filerne fra <http://getbootstrap.com/>, og inkludere dem på dit site (nødvendigt, hvis du skal arbejde uden internetforbindelse). Der er mange filer i den mappe du downloader. De interessante er dem du finder i "dist"-mappen. Du skal inkludere "bootstrap.min.css" i <head>-delen af dit site. "bootstrap.min.js" skal inkluderes lige før </body>-taget, der lukker bodyen.

B) Inkludere filerne fra et CDN (netforbindelse nødvendig under udvikling - netforbindelsen kan drille i ny og næ, så jeg anbefaler metode A):

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/
bootstrap.min.css">
<!-- Latest compiled and minified JavaScript -->
<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></
script>
```

Udover Bootstrap-filerne, skal jQuery også inkluderes (i hvert fald hvis der skal gøres brug af Bootstraps udvidede funktionaliteter). jQuery kan downloades fra <http://jquery.com/> eller inkluderes via jQuery-sitet: `<script src="//code.jquery.com/jquery.js"></script>`. jQuery skal inkluderes lige før "bootstrap.min.js"-filen.

Bootstrap bygger på HTML5 og CSS media queries, som kan drille lidt i tidlige versioner af IE. Hvis IE8 skal understøttes, skal html5shiv.js og responsejs inkluderes - de to filer kan downloades fra hhv. <https://code.google.com/p/html5shiv/> og [responsejs.com](http://responsejs.com). Inkluder dem f.eks. lige før din `</head>` lukkes, omkranset i en "IE-betingelse":

```
<!-- HTML5 shiv og and Respond.js includes -->
<!--[if lt IE 9]>
    <script src="incs/html5shiv.js"></script>
    <script src="incs/response.min.js"></script>
<![endif]-->
```

Samlet bliver resultatet nogenlunde som følgende med lokal installation af hele pibtøjet, alt efter hvordan du strukturerer dine mapper:

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

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Sitenavn . dk</title>

    <!-- Bootstrap CSS-includes -->
    <link rel="stylesheet" href="incs/bootstrap.min.css">

    <!-- HTML5 shiv og and Respond.js includes -->
    <!--[if lt IE 9]>
        <script src="incs/html5shiv.js"></script>
        <script src="incs/response.min.js"></script>
    <![endif]-->

  </head>
  <body>

    <h1>Halløjsa!</h1>

    <!-- jQuery & bootstrap-JS includes -->
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="incs/jquery-1.10.2.min.js"></script>
    <!-- Latest compiled and minified JavaScript -->
    <script src="incs/bootstrap.min.js"></script>

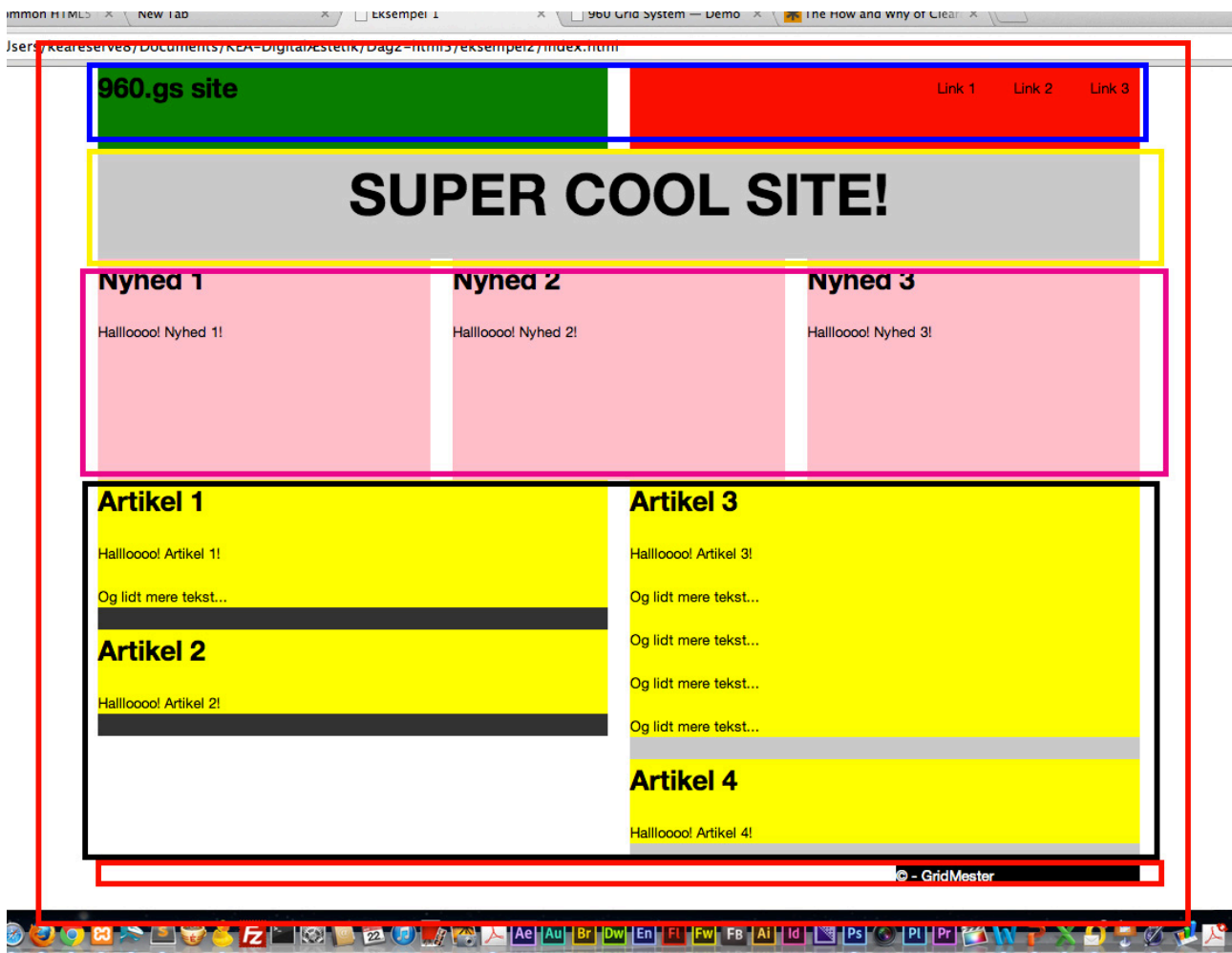
  </body>
</html>
```

# Layout med Grid og Scaffolding

## Grid systemet

Layout med html og CSS kan ofte være lidt drilskt. Derfor kan det være rigtig rart at bruge et grid-system, som hjælper én med koden til placering af elementerne på sitet.

Indtil mobil-bølgen kom, plejede vi at arbejde med sites, der var 960px bredde. Layoutdiagrammet var det vigtigste dokument til at hjælpe med opdelingen. Først delte man sitet op i rækker - altså horisontale sektioner af indhold. Dernæst delte man hver række op i kolonner. En række kunne med fordel deles op i 12 kolonner, som hver var 80px bredde - altså 960px i alt. Hvert indholdselement i en række, kunne så fylde 1, 2, 3.... til max 12 kolonner, dvs. 80, 160, 240.... 960px.



Nu hvor mobilene har taget en stor del af markedet, skal vi tænke et nyt lag ind i vores grids: Hvordan de håndterer forskellige skærmstørrelser. Bootstrap har et responsivt grid indbygget, som vi skal arbejde med. Det er mobile first - dvs. at man først laver den mobile løsning, og derefter tilpasser man sitet til de større enheder.

Grundlaget er det samme som før: Først og fremmest deler vi sitet op i rækker, og dernæst har vi 12 kolonner at dele hver række ind i. Nu er kolonnerne dog ikke nødvendigvis 80px bredde - de varierer i forhold til skærmbredden.

Der er 4 størrelser i bootstrap grid'et, som bruges til at dele en række op i kolonner, alt efter hvor stor den pågældende skærm er. Man behøver ikke at bruge alle 4 størrelser i sit layout - det gælder først og fremmest om at blive fortrolig med grid'et, derefter kan man eksperimentere.. De 4 forskellige grids har følgende bredder og breakpoints:

- **.col-xs-** : Definerer sitet, når vi er på en enhed, der er mindre end 768 px. XS-gridet passer til mobilenheder, og deler sitet op i kolonner, når vi er på enheder under 768px. Det kan give nogen meget små kolonner, når man er på en lille skærm, så arbejder man med xs-gridet, gælder det om at teste på mobilen/et meget smalt browservindue. XS-gridet bruges til layout på mobiler - mobile first! Bruger man udelukkende XS-gridet, kan det også give nogen meget brede kolonner på en større skærm.
- **.col-sm-** : Definerer sitet over 768px - container max 750px. SM-gridet er kollapset til at starte med, horisontalt layout over 768 px. Dvs. at deler vi hele vores site op i .col-sm- kolonner, kommer alle indholdselementer på vores site til at være i én kolonne under hinanden, når vi er under 768px. Når vi kommer derover, vil vi få flere kolonner, der vises ved siden af hinanden.
- **.col-md-** : Definerer sitet over 992px - container max 970px. Kollapset til at starte med, horisontalt layout over 992 px. Dvs. at deler vi hele vores site op i .col-md- kolonner, kommer alle indholdselementer på vores site til at være i én kolonne under hinanden, når vi er under 992px. Når vi kommer derover, vil vi få flere kolonner, der vises ved siden af hinanden.
- **.col-lg-** : Definerer sitet over 1200px - container max 1170px. Kollapset til at starte med, horisontalt layout over 1200px. Dvs. at deler vi hele vores site op i .col-md- kolonner, kommer alle indholdselementer på vores site til at være i én kolonne under hinanden, når vi er under 1200px. Når vi kommer derover, vil vi få flere kolonner, der vises ved siden af hinanden.

Flere informationer om gridet samt eksempler kan findes på <http://getbootstrap.com/css/>

## Scaffolding

Lad os starte med at anvende ét grid - så kan vi vende tilbage til at anvende flere grids på én gang senere. Vi anvender først MD-gridet, så vores site er "mobilt," hvis skærmen er under 992px, og stationært ved over 992px.

Når man skal konstruere wireframen på sit site, kaldes det scaffolding. Med Bootstrap-gridet skal man bruge en række HTML/CSS classes, der er indbygget i Bootstrap CSS'en. Først og fremmest skal alt indholdet i <body>en pakkes ind i en <div class="container">, for at sitet ikke er pakket helt tæt ind til kanterne på browseren (container class'en er faktisk ikke obligatorisk, men for os er den en hjælp mens vi lærer systemet at kende).

Hver gang man starter en ny række indhold i henhold til ens layoutdiagram, skal hele denne række pakkes ind i et html-element med classen row: <div class="row">

En <div class="row"> består af 12 kolonner, hvilket vil sige at man kan dele indholdselementerne i den pågældende <div class="row"> op i 12 kolonner. For at angive at et element har en bestemt bredde, lægges en <div class="col"> på elementet, for at angive at det består af en kolonne. Dernæst kan en yderligere detaljer vedr. class'en

påføres - f.eks. col-md-1, col-md-2, col-md-3... col-md-12, for at angive hvor mange af de 12 kolonner i rækken, det pågældende element fylder:

```
<div class="col-md-1"> <div class="col-md-2"> <div class="col-  
md-3">... <div class="col-md-12">
```

Det vigtige i forbindelse med at tilføje "cols" i en row er at det totale antal cols ikke må overstige 12. Man kan altså kombinere 12 elementer med col-md-1 classes på (men ikke 13), 2 elementer med col-md-6 classes på, 3 elementer med col-md-4 classes på osv.

## Eksempel på scaffolding med col-md-gridet

```
1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width,
6   initial-scale=1.0">
7   <!-- viewport for korrekt skalering på tværs af
8   medieenheder -->
9
10  <title>Bootstrap-site</title>
11
12  <link href="css/bootstrap.min.css" rel="stylesheet">
13  <link href="css/custom.css" rel="stylesheet">
14
15  <!-- Respond.js for kompatibilitet med gamle IE-
16  browsere - https://github.com/scottjehl/Respond -->
17  <script src="js/respond.min.js"></script>
18 </head>
19 <body>
20   <div class="container">
21     <header class="row">
22       <section>
23         navigation
24       </section>
25     </header>
26
27     <div class="row">
28       <section class="col-md-3 offset-md-9">
29         <p>Garagefilm er fuld af sjov og
30         ballade...</p>
31       </section>
32
33     </div>
34
35     <div class="row">
36       <h1>Seneste nyt</h1>
37       <article class="col-md-4">
38         <h1>Nyhed 1</h1>
39         <p>Nyhed 1 er kort...</p>
40       </article>
41       <article class="col-md-4">
```

```
38     <h1>Nyhed 2</h1>
39     <p>Nyhed 2 er kort...</p>
40 </article>
41 <article class="col-md-4">
42     <h1>Nyhed 3</h1>
43     <p>Nyhed 3 er kort...</p>
44 </article>
45 </div>
46
47 <div class="row">
48     <h1>Produktioner</h1>
49     <article class="col-md-3">
50         <p>Produktioner 1</p>
51     </article>
52     <article class="col-md-3">
53         <p>Produktioner 2</p>
54     </article>
55     <article class="col-md-3">
56         <p>Produktioner 3</p>
57     </article>
58     <article class="col-md-3">
59         <p>Produktioner 4</p>
60     </article>
61 </div>
62
63 <div class="row">
64 <h1>Socialt</h1>
65     <section class="col-md-12">
66         <p>Social nødhjælp</p>
67     </section>
68 </div>
69
70 <div class="row">
71 <h1>Kontakt</h1>
72     <section class="col-md-6">
73         Kontakt
74     </section>
75 </div>
76
77 <footer class="row">
78     footer
```



```
80
81     </div><!-- end container -->
82     <!-- Javascript -->
83     <!-- jQuery v 1.10 - bedre end v. 2 når det kommer til
84     IE8 -->
85     <script src="http://code.jquery.com/jquery-
86     latest.min.js"></script>
87     <!-- Tilføj bootstrap js EFTER jQuery -->
88     <script src="js/bootstrap.min.js"></script>
89 </body>
90 </html>
```

### Offsetting

Vil man forskyde en kolonne, kan man bruge de indbyggede offset-classes. Har man en kolonne med en class col-md-3, f.eks., fylder den 3 kolonner. Den kan så forskydes fra venstre side med et bestemt antal kolonner ved at tilføje col-md-offset-X classen, som i eksemplet nedenfor. Der føjes altså to classes til det pågældende element:

```
<div class="col-md-3 col-md-offset-3">
```

### Kombination af grids

Når vi kun arbejder med ét grid, f.eks. col-md-grid'et, bliver elementerne på vores site placeret i vertikal orden, når browseren er mindre end 992px. Det kan godt give nogen ret lange linielængder, når vores skærm er lige under 992px bred, og vi dermed er i ét-kolonnet layout.

Derfor kan man kombinere to grids - altså f.eks. col-md og col-sm gridene. Så kan vi give et element den regel at det skal fylde f.eks. 4 kolonner i col-md-gridet, og 6 kolonner i col-sm-gridet. Med andre ord: Elementet fylder 1/3-del af bredden på skærmen, hvis vi har en browser, der er over 992px bred, og elementet fylder 1/2-delen, hvis vi er på en skærm, der er under 992px bred. Det kan gøres som følgende:

```
<div class="col-md-4 col-sm-6">
```

På den måde, kan vi sørge for at indholdet på sitet reorganiseres, alt efter hvilken skærmstørrelse, der er tale om. Det betyder at der skal mange classes på alle elementer, for at gøre vores layout fleksibelt!

Husk på at xs-gridet er horisontalt layoutet uanset skærmstørrelse - xs-gridet kan altså bruges til at designe den mobile oplevelse af sitet.

### Show og hide classes

Ud over grid-systemet, har Bootstrap også en række show og hide-classes. Det er kort sagt classes vi kan lægge på vores elementer, så de udelukkende vises eller skjules når vi har en bestemt skærmstørrelse:

- visible-xs - vises når vi er på en skærm under 768px
- visible-sm - vises når vi er på en skærm mellem 768px og 992px

- visible-md - vises når vi er på en skærm mellem 992px og 1200px
- visible-lg - vises når vi er på en skærm over 1200px
- ... og... hidden-xs, hidden-sm, hidden-md, hidden-lg

Det er vigtigt at tænke på at elementerne stadig skal downloades, selvom der er en hide-class på det - så lad være med at lægge kæmpe billeder ind, og giv dem en hide-class på telefonen. Det kan gøre sitet ret tungt på mobilen!

### Ændring af kolonne-ordenen

Når sitet er i mobil-mode, organiseres alle elementer i henhold til den rækkefølge, de kommer i i html-dokumentet. Elementernes rækkefølge bør være den rigtige, når vi ser sitet på en mobil-enhed.

Vil vi gerne omorganisere elementernes rækkefølge, kan vi bruge push-og pull classes på de forskellige elementer.

Har vi f.eks. en kolonne med to elementer, når vi er i md-gridet, vil de blive vist i én kolonne-layout, når vi er på mobilen.

```
<div class="col-md-6">Første div</div>
<div class="col-md-6">Anden div</div>
```

Vil vi gerne bytte om på rækkefølgen af elementerne, når vi er på en stor skærm, kan vi give det første element en "push" class, for at skubbe det til højre, og det andet element en "pull" class for at trække det mod venstre:

```
<div class="col-md-6 col-md-push-6">Første div (vises sidst)</div>
<div class="col-md-6 col-md-pull-6">Anden div (vises først)</div>
```

### Nesting/indlejring af rows i kolonner

En ny row kan oprettes i et andet html-element med en col-bredde på - rows kan nestes/indlejres i andre html-elementer. Det giver muligheden for at lave avancerede layouts. Eksempelvis:

```
<div class="col-md-6">
Hoveddiv
<div class="row">
    <div class="col-md-6">Underelement i nested row</div>
    <div class="col-md-6">Underelement i nested row</div>
</div>
</div>
```

### Jumbotron/hero-unit

En hero-unit er et stort element, typisk på forsiden af ens website, der prøver at tiltrække opmærksomhed. I bootstrap lingo hedder den jumbotron. Jumbotronen fylder alle 12 kolonner af websitet - dvs. fuld bredde. Den kan oprettes ved at tilføje klassen jumbotron til et element. Læg mærke til at jumbotron class'en erstatter en <div class="row">:

```
<div class="jumbotron">
    Jumbotron
</div>
```

## Div class clearfix

Bootstrap er bygget med floats, og derfor kan det indimellem være nødvendigt at klare sine floats, for at elementerne ikke kolliderer (det kan du finde mange artikler om, hvis du søger "clear your floats" på Google). Bootstrap har en indbygget clear klasse, som tilføjes lige før den container, der skal klare sig. Eksempelvis:

```
<div>
  <p>Masser af indhold her...</p>
  <div class="clearfix"></div>
</div>
```

## Indbygget CSS/Styling

Twitter bootstrap kommer med en masse standard styling, der er meget minimalistisk... Det er kort sagt en masse classes vi kan tilføje vores html-elementer, for at style dem. Her en kort gennemgang af nogen af de vigtigere - se mere på <http://getbootstrap.com/css/>

### Generelt:

class="lead" - lidt større/fremhævet

Tekst kan fremhæves i html med: <em>text</em>

Citater/uddrag mm. kan indrykkes med: <blockquote>Tekst</blockquote>

Tekst kan reduceres i styrke/størrelse med <small>Tekst</small>

Elementer kan floates til højre og venstre med class="pull-right" og "pull-left" - men efterfølgende elementer skal klare sig (clear your floats!), hvis de først skal komme efter elementet.

class="unstyled" kan bruges til at lave lister uden formattering

### Knapper:

Der følger en hulens masse formaterede knapper med i bootstrap CSS'en - se <http://getbootstrap.com/css/#buttons>:

class="btn" indikerer at et link er en knap - den gør ikke meget i sig selv, og står derfor sjældent alene. Den følges af en yderligere class:

class="btn btn-default" som giver en noget kedeligt stilet knap...

class="btn btn-primary" kan bruges til at gøre knappen til en (blå) "primær-knap"

class="btn btn-success" kan bruges til at lave en (grøn) succes knap

class="btn btn-warning" kan bruges til at lave en (gul/orange) advarsels-knap

class="btn btn-danger" kan bruges til at lave en (rød) fare-knap

class="btn btn-inverse" kan bruges til at lave en negativt farvet (sort) knap

(farverne kan overrides)

class="btn-lg" "btn-md" "btn-sm" og "btn-xs" justerer størrelsen af knapperne

class="btn-block" kan bruges til at gøre knappen lige så bred, som det container-element, den ligger i

Og så kan man definere sine egne styles til knapperne...

### Billeder:

Som udgangspunkt er billederne i Bootstrap ikke responsive (de tilpasser sig ikke i størrelse). Men funktionaliteten er indbygget, der skal blot tilføjes et par classes til <img> elementerne:

class="img-responsive" gør billederne responsive

class="img-circle" klipper billedet til en cirkel

class="img-rounded" giver billedet rundede hjørner

class="img-thumbnail" giver en lille hvid kant på billedet  
pull-right og pull-left class'erne kan bruges til at floate billeder hhv. til højre og venstre.

### **Ikoner:**

Bootstrap inkluderer ikoner fra Glyphicons (dem skal der normalt betales for, men Twitter har lavet en god ordning med dem...).

<http://getbootstrap.com/components/#glyphicons>

Ikonerne inkluderes ved eksempelvis: `<span class="glyphicon glyphicon-plane"></span>`  
for et fly-ikon.

Glyphiconerne er tilføjet som en font i Bootstrap, hvilket betyder at de kan skaleres op og ned efter behov, og man kan ændre farven som det passer én (via CSS).

### **Overriding CSS:**

To måder at gøre det på: Customize bootstrap.

Customize bootstrap: <http://getbootstrap.com/customize/> og tryk download og installer på dit site. Det kan dog drille lidt, når man installerer opdateringer, da de vil overskrive de filer, der er blevet ændret...

Derfor kan man lave et custom stylesheet. Det er egentlig blot at lave et stylesheet, der hedder custom.css, og inkludere det i ens html-fil EFTER bootstrap stylesheeten.

## **Navigation**

Navigation er noget af det vigtigste på sitet, og nu hvor vi har fået et grundlæggende kendskab til bootstrap, herunder grid'et og CSS'en, er det på tide at arbejde videre med navigationselementerne - dem har bootstrap nemlig rigtig mange funktioniteter til.

Den simpleste måde at komme i gang på er at tilføje et `<nav>` element og en ul med classen "nav."

```
<nav>
  <ul class="nav">
    <li><a href="#hjem" alt="">Hjem</a></li>
    <li><a href="#nyheder" alt="">Nyheder</a></li>
    <li><a href="#produktioner" alt="">Produktioner</a></li>
    <li><a href="#socialt" alt="">Socialt</a></li>
    <li><a href="#kontakt" alt="">Kontakt</a></li>
  </ul>
</nav>
```

I ovenstående menu lægger jeg op til et single page layout, hvor hver link går til en sektion på den samme side, i stedet for til en ny html-side. Dvs. at href'en for hver link kører til et element, der id'et #et\_eller\_andet på sig.

I følgende menu er der blevet tilføjet et par classes: nav-tabs sørger for at gøre `<li>` elementerne til tabs (godt til en horisontal menu) og nav-stacked gør tabsene til en vertikal menu. class="active" henfører til den side vi er på (aktuel side highlighting).

```

<nav>
  <ul class="nav nav-tabs nav-stacked">
    <li class="active"><a href="#hjem" alt="">Hjem</a></li>
    <li><a href="#nyheder" alt="">Nyheder</a></li>
    <li><a href="#produktioner" alt="">Produktioner</a></li>
    <li><a href="#socialt" alt="">Socialt</a></li>
    <li><a href="#kontakt" alt="">Kontakt</a></li>
  </ul>
</nav>

```

Andre alternativer:

`<ul class="nav nav-pills">` for pill-navigation

`<ul class="nav nav-pills nav-justified">` sørger for at gøre menuen 100% af container elementet

## Responsiv navigation

Bootstrap har en række jQuery-baserede plugins, som kan bruges for at gøre vores navigationsmenu mere responsive (så de kollapser ved små skærmstørrelser). Først og fremmest skal vi sørge for at jQuery og Bootstrap.min.js er inkluderet. Det skulle de gerne være!

To HTML5 attributter bliver brugt meget: Data-toggle og data-target. Data-toggle bruges til at angive hvilket element, der skal sørge for at der sker noget (En Event Listener). Data-target angiver hvilket element handlingen skal udføres på (Event target).

```

<nav class="navbar navbar-default" role="navigation">
  <div class="navbar-header">
    <button type="button" class="navbar-toggle" data-toggle="collapse" data-
      target="#collapse">
    <span class="sr-only">Toggle Navigation <!-- Kun til Screen Readers
      accessibility -->
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
  </div>
  <div class="collapse navbar-collapse" id="collapse">
    <ul class="nav navbar-nav">
      <li class="active"><a href="#hjem" alt="">Hjem</a></li>
      <li><a href="#nyheder" alt="">Nyheder</a></li>
      <li><a href="#produktioner" alt="">Produktioner</a></li>
      <li><a href="#socialt" alt="">Socialt</a></li>
      <li><a href="#kontakt" alt="">Kontakt</a></li>
    </ul>
  </div>
</nav>

```

Der er altså en række nye classes at bruge, når menuen skal laves responsiv. Alt det i navbar-header bruges kort sagt til at vise den kollapsede menu. De tre `<span class="icon-bar">` er glyphiconer, i dette tilfælde tre vandrette streger, som ofte bruges til at indikere at det er en menu (er det ved at blive en konvention på mobilsites?). Alt det i div'en med id'et collapse, er det der vises, hvis vi er på en stor skærm.

Navigationsmenuen kan styles yderligere, ved at bruge en række andre/yderligere classes: Enten dem vi allerede har kigget på, andre fra bootstrap-dokumentationen, eller egne classes. Nogen gode classes at kende er:

`<nav class="navbar navbar-default navbar-inverse">`: Inverterer farvepaletten

`<nav class="navbar navbar-default navbar-fixed-top">`: Positionerer menuen fast i toppen af sitet - 70 px padding-top på body'en bliver nødvendigt, da menuen overlapper andet indhold.

`<nav class="navbar navbar-default navbar-fixed-bottom">`: Som top, bare i bunden af sitet i stedet for i toppen

`<nav class="navbar navbar-default navbar-static-top">`: Laver en navigationsbar i toppen af sitet, som scroller med indholdet.

## Dropdowns

Med Bootstrap og jQuery kan vi lave dropdowns, der også virker på mobilenheder. Eksempelvis, fortsat fra sidste nav-menu, er det interessante at se på den indskudte `<ul>`:

...

```
</div>
<div class="collapse navbar-collapse" id="collapse">
  <ul class="nav navbar-nav">
    <li class="active"><a href="#hjem" alt="">Hjem</a></li>
    <li><a href="#nyheder" alt="">Nyheder</a></li>
    <li class="dropdown"><a href="#" data-toggle="dropdown"
      alt="">Produktioner <span class="caret"></span></a>
      <ul class="dropdown-menu">
        <li><a href="#produktioner" alt="">Reklamer</a></li>
        <li><a href="#produktioner" alt="">Kortfilm</a></li>
      </ul>
    </li>
    <li><a href="#socialt" alt="">Socialt</a></li>
    <li><a href="#kontakt" alt="">Kontakt</a></li>
  </ul>
```

...

Det samme kan gøres med pill-navigation: `<ul class="nav nav-pills">` er den eneste forskel...

## OBS! vedr. menuer og one-page layouts

Menuerne i Bootstrap kan i bund og grund styles, som det passer én. Men det kan indimellem være lidt tricky at finde ud af helt nøjagtigt, hvordan man skal gøre det.

Vil man f.eks. lave et one-page layout med en fast placeret topmenu, som kan folde sig ud, når man er på en mobil-enhed, skal menuen kollapse igen, når man har trykket på et link. Det skal man bruge et lille jQuery script til - noget lignende:

```
<!-- Kollaps menu, når der trykkes på et link-->
    $(' .navbar a').click(function() {
        var navbar_toggle = $(' .navbar-toggle');
        if (navbar_toggle.is(':visible')) {
            navbar_toggle.trigger('click');
        }
    });
```

Det skyldes at menuen som udgangspunkt bliver foldet ud, og bevares åben, til der bliver klikket på et link, eller menuen lukkes igen. Det behøves ikke, hvis man lav et site med flere sider, for så vil menuen reloades, når den nye side reloades. Men det sker ikke med et one-page layout, så her bliver man nødt til at tvinge menuen til at kollapse, når der trykkes på et link.

Se yderligere informationer om bootstrap navigation på <http://getbootstrap.com/components/#nav> og <http://getbootstrap.com/components/#navbar>.

## Customizing bootstrap

En nem genvej, når man skal til at customize sit projekt, er at gå ind på customize fanen, for at indsætte en række variable. <http://getbootstrap.com/customize/>

Her kan man bl.a. sætte sitets farvepalet, man kan ændre på @grid-float-breakpoint, for at justere hvornår ens navigationsmenu kollapse, man kan ændre på border-radius'es, farver for navigationsbaren mm.

Bootstrap er skrevet i LESS, dvs. precompilet CSS, og når man customizer bootstrap, får man bootstrap til at generere en ny udgave af Bootstrap. Det kan være en praktisk/hurtig måde at komme i gang på, men den kan også være lidt drilsk. Det er ikke muligt at gemme en profil for et website, og derfor skal man skrive ned, hvordan man har customizet sit site, hvis man vil ind og lave små ændringer i det, fordi alle tidligere ændringer skal testes ind igen. Derfor kan det være nemmere at lave overrides med CSS.