

Repetition

Idag ska vi repetera

Förra veckan kom det in önskemål på att repetera vissa av de saker vi gått igenom tidigare:

- CSS Selectors
- Media queries
- Bootstrap
- Flexbox (imorgon)

Planering

Vi gör därför några mindre ändringar i planeringen för de kvarvarande veckorna:

Vecka 41 - Repetition, Grid

Vecka 42 - Grid, Animationer, Media, Nyheter i CSS

Vecka 43 - Repetition + Tentamen

CSS selectors - repetition

Selector	Exempel	Förklaring
element	<code>p { ... }</code>	Väljer alla <code><p></code> -element
id	<code>#main { }</code>	Ett element med <i>id="main"</i>
class	<code>.highlight</code>	Alla element med <i>class="highlight"</i>
attribute	<code>input[type="email"]</code>	Alla input element med attributet <code>type="email"</code>
pseudo-class	<code>p:hover</code>	Alla p taggar som man har muspekaren över

Exempel

CSS selectors - repetition

Selector	Syntax	Exempel	Förklaring
Gruppering	A, B	p, h1 {}	Applicerar gemensam styling på alla p och h1 element
Descendant	A B	header p	Applicerar styling på alla p element inne i en header
Child	A > B	ul > li	Applicerar styling på alla li element som finns direkt under ett ul element

[Exempel](#)

Fler CSS selectors

Selector	Syntax	Exempel	Förklaring
General sibling	A ~ B	img ~ p	Alla <p>-element som kommer efter ett -element
Adjacent sibling	A + B	p + p	Alla <p>-element som kommer direkt efter ett annat <p>-element
First child	A:first-child	li:first-child	Alla -element som är det första barnet i en container
Last child	A:last-child	img:last-child	Alla -element som är det sista barnet i en container

Exempel

CSS Selectors Specificity

I vilken ordning evalueras CSS Selectors?

Finns en mycket bra artikel om det på MDN:

<https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity>

CSS Selectors

Minst prio

1. Ett element
2. Flera element
3. Klass
4. Element + klass
5. Flera klasser
6. Id
7. Klass + id
8. style-attribute
9. !important

Högst Prio

```
p { ... }  
div p { ... }  
.x { ... }  
div .a { ... }  
.x.y { ... }  
#myid { ... }  
.x #myid { ... }  
style="..."  
p !important
```


CSS Selectors Specificity

Eller om ni föredrar att läsa om i Star Wars-termer:

https://stuffandnonsense.co.uk/archives/css_specificity_wars.html



a

1 x element selector

Sith power: 0,0,1



p a

2 x element selectors

Sith power: 0,0,2



.foo

1 x class selector *

Sith power: 0,1,0



a.foo

1 x element selector
1 x class selector

Sith power: 0,1,1

CSS Selectors

I stora drag kan man säga att ju mer specifika vi är, ju större prioritet har vår regel.

Exempelvis går

```
.myclass p { font-size: 12px; }
```

före

```
p { font-size: 12px; }
```

Eftersom vi är mer *specifika* i den översta regeln

CSS Selectors

I vilken ordning evalueras css? Om vi t.ex. har ett element som ser ut så här:

```
<div class="wrapper">
  <div id="myid" class="myclass">
    Tjena
  </div>
</div>
```

```
.myclass {
  color: green;
}

.myclass {
  color: blue;
}
```

Vilken textfärg får elementen?

[JSFiddle](#)

CSS Selectors

Vad händer i det här läget?

```
<div class="wrapper">
  <div id="myid" class="myclass">
    Tjena
  </div>
</div>
```

```
#myid {
  color: green;
}

.myclass {
  color: blue;
}
```

Vilken textfärg får elementen?

[JSFiddle](#)

CSS Selectors

Vad händer i det här läget?

```
<div class="wrapper">
  <div id="myid" class="myclass" style="color: yellow">
    Tjena
  </div>
</div>
```

```
#myid {
  color: green;
}

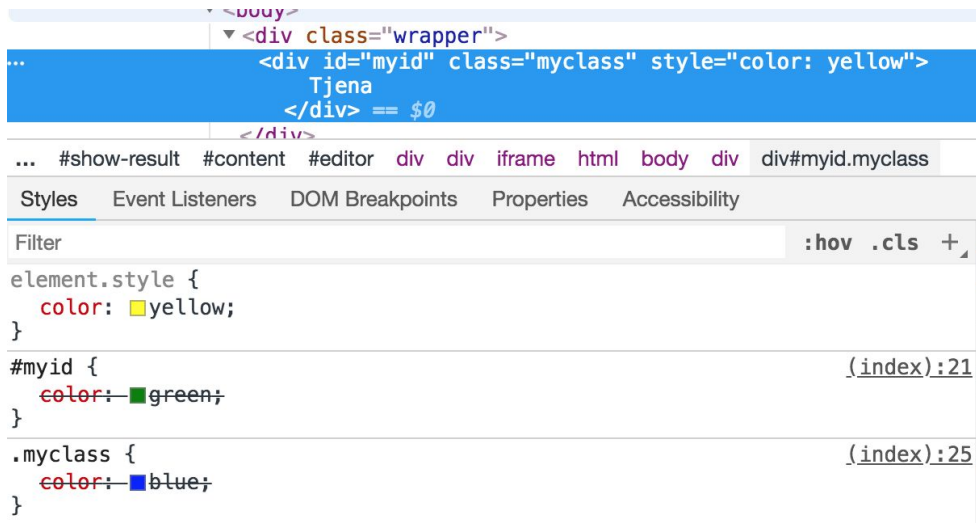
.myclass {
  color: blue;
}
```

Vilken textfärg får elementen?

[JSFiddle](#)

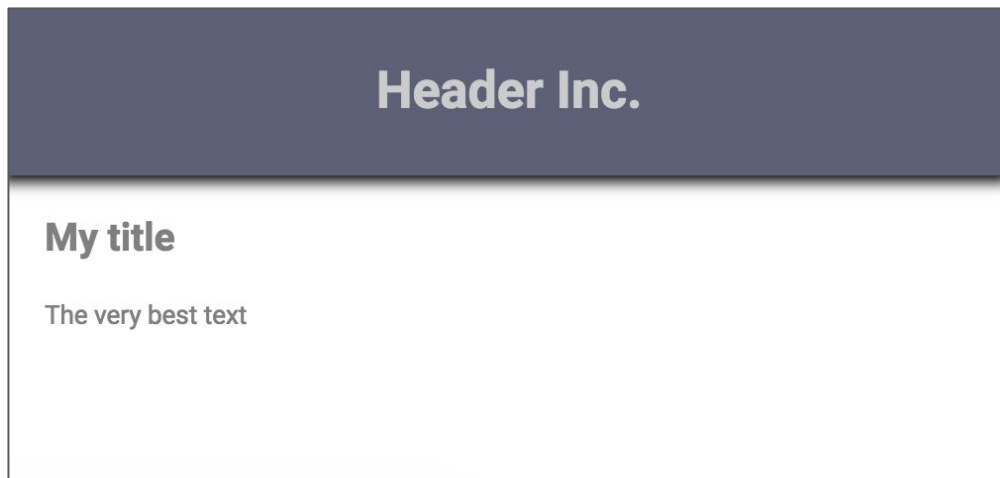
Felsökning

Om vi märker att en CSS-regel som vi skrivit inte verkar fungera kan vi undersöka med hjälp av Dev Tools:



CSS Selectors - Exempel

Vi har fått i uppgift att göra ett par förändringar på en sida:



1. Paragraftexten ska vara svart istället för grå.
2. Header ska ha större padding (20 px)
3. “Header Inc.”-texten ska vara vit istället för grå.

[Exempel](#)

CSS Selectors - Exempel

Vi inser dock att CSS-reglerna för sidan är stylade med hjälp av en CSS-fil från en tredje part som vi inte kan ändra på. Vi behöver alltså skriva över en del av reglerna.

```
/** CSS från en annan fil som vi inte får/kan ändra i **/  
  
body {  
  margin: 0;  
}  
  
.header-container {  
  padding: 10px;  
  text-align: center;  
  background: #5e5f79;  
  box-shadow: -2px 3px 10px;  
}
```


Paragrafen

Vi börjar med paragrafen:

```
p, h2 {  
  margin: 20px;  
  color: #808080;  
}
```

Här kan vi vara mer specifika genom att lägga till klassen från p-taggens förälder först i regeln:

```
.page-content p {  
  color: #000;  
}
```

[Exempel](#)

Headern

Headern (som ska ha padding 20px) har följande styling.

```
.header-container {  
  padding: 10px;  
  text-align: center;  
  background: #5e5f79;  
  box-shadow: -2px 3px 10px;  
}
```

```
<header>  
  <div class="header-container">  
    <h1 id="header-title">  
      Header Inc.  
    </h1>  
  </div>  
</header>
```

Här kan vi vara mer specifika genom att t.ex. lägga till elementet först i regeln:

```
div.header-container {  
  padding: 20px;  
}
```

[Exempel](#)

Titeln

Slutligen har vi titeln, som är stylad med hjälp av !important.

```
#header-title {  
  color: #ccc !important;  
}
```

```
<header>  
  <div class="header-container">  
    <h1 id="header-title">  
      Header Inc.  
    </h1>  
  </div>  
</header>
```

Titeln

Här blir vi tvungna att också använda oss av `!important`, men vi får helt enkelt vara mer specifika.

```
.header-container #header-title {  
  color: #fff !important;  
}
```

[Exempel](#)

Media queries

Används för att ändra ett elements egenskaper beroende på egenskaper hos användarens enhet.

Exempel:

- width
- height
- print
- orientation

Media Types

- all
- screen
- print

Sätts ingen media query är "all" default

Media Queries

Ett par olika exempel:

```
@media (min-width: 500px) { ... }
```

```
@media (min-width: 500px) and (max-width: 600px) { ... }
```

```
@media (min-width: 500px), (min-height: 500px) { ... }
```

[Exempel](#)

Media queries - exempel

Titel och bild

[JSFiddle](#)

Hur kan vi göra texten mindre och att bilden fylls ut på mindre skärmar?

[JSFiddle](#)

Bootstrap - Backstreet Boys

Vi ska skapa fansida för Backstreet Boys i tre olika storlekar:

Backstreet Boys

The Unofficial Fan Page - Backstreet's back, alright!



Nick



Brian



A.J.



Kevin



Howard

Backstreet Boys

The Unofficial Fan Page - Backstreet's back, alright!



Nick



Brian



A.J.



Kevin



Backstreet Boys

The Unofficial Fan Page - Backstreet's back, alright!



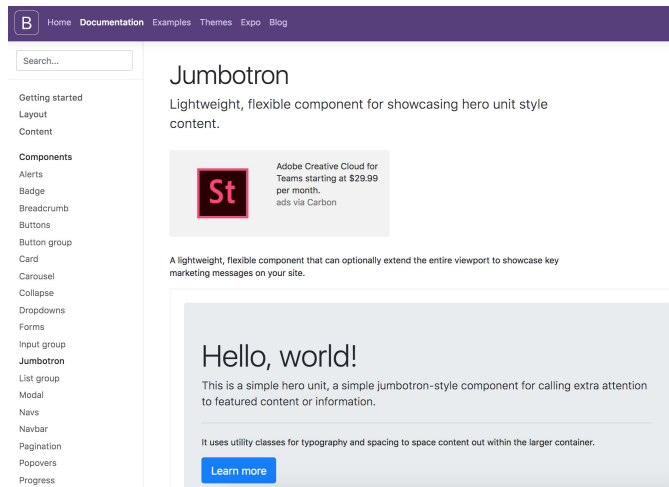
Nick



Vi börjar med en Jumbotron

Precis som tidigare utgår vi från dokumentation:

<https://getbootstrap.com/docs/4.1/components/jumbotron/>



The screenshot shows the Bootstrap 4.1 Jumbotron documentation page. The page has a purple header with a navigation bar containing links: Home, Documentation, Examples, Themes, Expo, and Blog. Below the header is a search bar and a sidebar with a list of components. The main content area is titled 'Jumbotron' and describes it as a 'Lightweight, flexible component for showcasing hero unit style content.' It features a code example of a Jumbotron with a 'Hello, world!' message and a 'Learn more' button. The sidebar lists various components including Alerts, Badge, Breadcrumb, Buttons, Button group, Card, Carousel, Collapse, Dropdowns, Forms, Input group, Jumbotron, List group, Modal, Navs, Navbar, Pagination, Popovers, and Progress.

Search...

Getting started
Layout
Content

Components

- Alerts
- Badge
- Breadcrumb
- Buttons
- Button group
- Card
- Carousel
- Collapse
- Dropdowns
- Forms
- Input group
- Jumbotron**
- List group
- Modal
- Navs
- Navbar
- Pagination
- Popovers
- Progress

Jumbotron

Lightweight, flexible component for showcasing hero unit style content.

St Adobe Creative Cloud for Teams starting at \$29.99 per month, ads via Carbon

A lightweight, flexible component that can optionally extend the entire viewport to showcase key marketing messages on your site.

Hello, world!

This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.

It uses utility classes for typography and spacing to space content out within the larger container.

[Learn more](#)

Jumbotron

Vi anpassar den till vårt exempel. Bland annat vill vi ha en vit bakgrund, vilket vi uppnår med `bg-white`:

```
<div class="jumbotron bg-white jumbotron-fluid">
  <div class="container">
    <h1 class="display-4">Backstreet Boys</h1>
    <p class="lead">The Unofficial Fan Page - Backstreet's back, alright!</p>
  </div>
</div>
```

Backstreet Boys

The Unofficial Fan Page - Backstreet's back, alright!

Resultat

Layout

Layouten består av tre storlekar. Vi börjar med den allra minsta.

I den ska varje bild fylla upp en egen rad så vi sätter `col-12` som utgångspunkt.

```
<div class="container">
  <div class="row">
    <div class="col-12">Nick</div>
    <div class="col-12">Brian</div>
    <div class="col-12">A.J.</div>
    <div class="col-12">Kevin</div>
    <div class="col-12">Brian</div>
  </div>
</div>
```

Resultat

Backstreet Boys

The Unofficial Fan Page - Backstreet's back, alright!



Nick



Layout

Vi fortsätter med nästa storlek, dvs den mittersta.

Vi ser att alla ska ta upp hälften av utrymmet, så vi använder col-sm-6.

```
<div class="container">
  <div class="row justify-content-center">
    <div class="col-12 col-sm-6">Nick</div>
    <div class="col-12 col-sm-6">Brian</div>
    <div class="col-12 col-sm-6">A.J.</div>
    <div class="col-12 col-sm-6">Kevin</div>
    <div class="col-12 col-sm-6">Brian</div>
  </div>
</div>
```

Resultat

Backstreet Boys

The Unofficial Fan Page - Backstreet's back, alright!



Nick



Brian



A.J.



Kevin



Layout

Slutligen har vi den sista layouten, dvs den största.

Här ser vi att vi ska ha två kolumner som ligger över de sista tre kolumnerna.

Eftersom de två raderna med bilder aldrig ska ligga på samma rad väljer vi att lägga dem i olika rader (. row)

Dessutom lägger vi till .col-md-4 på alla eftersom de tar upp en $\frac{1}{3}$ av utrymmet

Backstreet Boys

The Unofficial Fan Page - Backstreet's back, alright!



Nick



Brian



A.J.



Kevin



Howard

Layout

Vi får då följande resultat:

```
<div class="container">
  <div class="row justify-content-center">
    <div class="col-12 col-sm-6 col-md-4">Nick</div>
    <div class="col-12 col-sm-6 col-md-4">Brian</div>
  </div>
  <div class="row justify-content-center">
    <div class="col-12 col-sm-6 col-md-4">A.J.</div>
    <div class="col-12 col-sm-6 col-md-4">Kevin</div>
    <div class="col-12 col-sm-6 col-md-4">Brian</div>
  </div>
</div>
```

Resultat

Layout

Sedan lägger vi in bilderna:

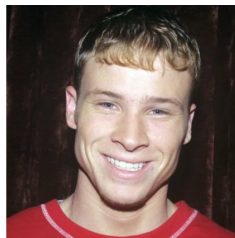
```
<div class="container">
  <div class="row justify-content-center">
    <div class="col-12 col-sm-6 col-md-4">
      
      <div class="name">Nick</div>
    </div>
    <div class="col-12 col-sm-6 col-md-4">
      
      <div class="name">Brian</div>
    </div>
  </div>
  <div class="row justify-content-center">
    <div class="col-12 col-sm-6 col-md-4">
      
      <div class="name">A 1</div>
    </div>
  </div>
</div>
```


Layout

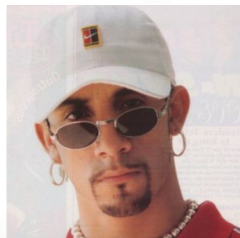
Vi ser nu att bilderna ligger korrekt enligt den layout vi ville ha.



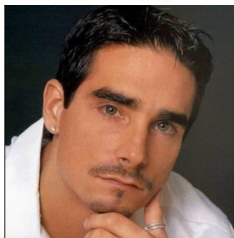
Nick



Brian



A.J.



Kevin



Howard

Resultat

Layout

Slutligen lägger vi på den sista stilingen för att göra bilderna runda, lägga på skugga och dylikt:

```
img {  
  width: 100%;  
  border-radius: 50%;  
  box-shadow: 0 .5rem 1rem rgba(0,0,0,.6);  
}  
  
.row > div {  
  text-align: center;  
}  
  
.name {  
  margin: 20px;  
}
```

Slutligt resultat

Nu är vi klara!



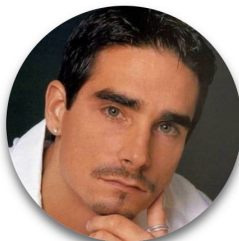
Nick



Brian



A.J.



Kevin



Howard

Resultat