

Flexbox

Flexbox

- En modell för att bygga layouter med CSS
- Strukturera innehåll i en container
- Responsivt

Flexbox: eksempel

```
<div class="container">  
  <div class="item"></div>  
  <div class="item"></div>  
  <div class="item"></div>  
</div>
```

```
.container {}  
  
.item {  
  background: #ccc;  
  margin: 10px;  
  height: 100px;  
}
```

Flexbox: exempel

```
.container {  
    display: flex;  
}
```

```
.item {  
    background: #ccc;  
    margin: 10px;  
    height: 100px;  
    flex: 1;  
}
```

Flexbox

`display: flex`

- Sätts på en container
- Innehållet i containern blir **flex-items**

`flex: <value>`

- Sätts på **flex-item**-element
- Styr hur elementen förhåller sig till varandra

container



items



Flexbox - flex: <value>

flex: <value>

- En kortare regel för
 - flex-grow (default 0)
 - flex-shrink (default 1)
 - flex-basis (default auto)

flex-grow

Bestämmer hur mycket ett element växer i förhållande till andra element

flex-shrink

Bestämmer hur mycket ett element krymper i förhållande till andra element

flex-basis

Bestämmer hur stort ett flex-element ska vara.

Flex - shrink vs grow

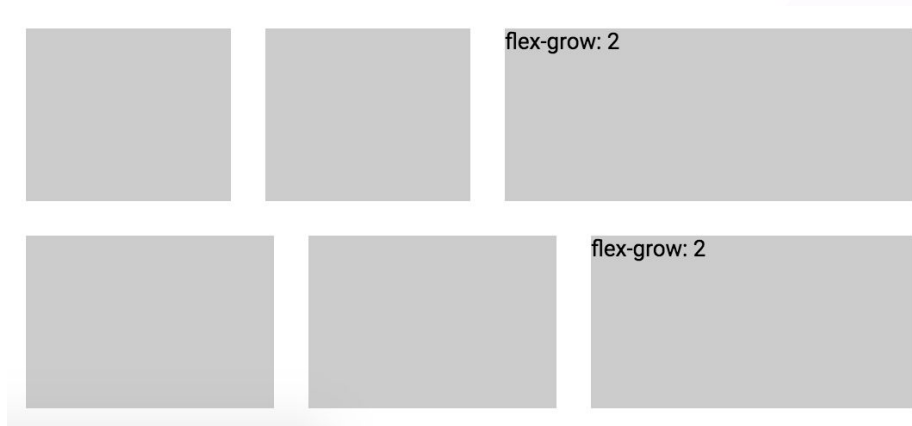
När ett element ska växa eller krympa beror på **flex-basis**

- Om elementets storlek är **mindre än flex-basis** ändras storleken enligt **flex-shrink**.
- Om elementets storlek är **större än flex-basis** ändras storleken enligt **flex-grow**.

flex-basis: auto och flex-basis:0

flex-basis: auto är standardvärdet på ett flex-item. Det betyder att flex-basis-värdet bestäms av storleken på elementet, t.ex. width.

flex-basis: 0 betyder att det inte finns någon initial storlek att förhålla sig till.



Main axis & Cross axis

Det finns två axlar i en flexbox layout: main och cross

- Flex-elementen lägger sig på main-axeln.
- Riktningen på main-axeln bestäms av `flex-direction`:
 - `flex-direction: row` innebär att main-axeln ligger horisontellt på en rad.
 - `flex-direction: column` innebär main-axeln ligger lodrätt i en kolumn
- Cross-axeln går i 90-graders vinkel från main-axeln

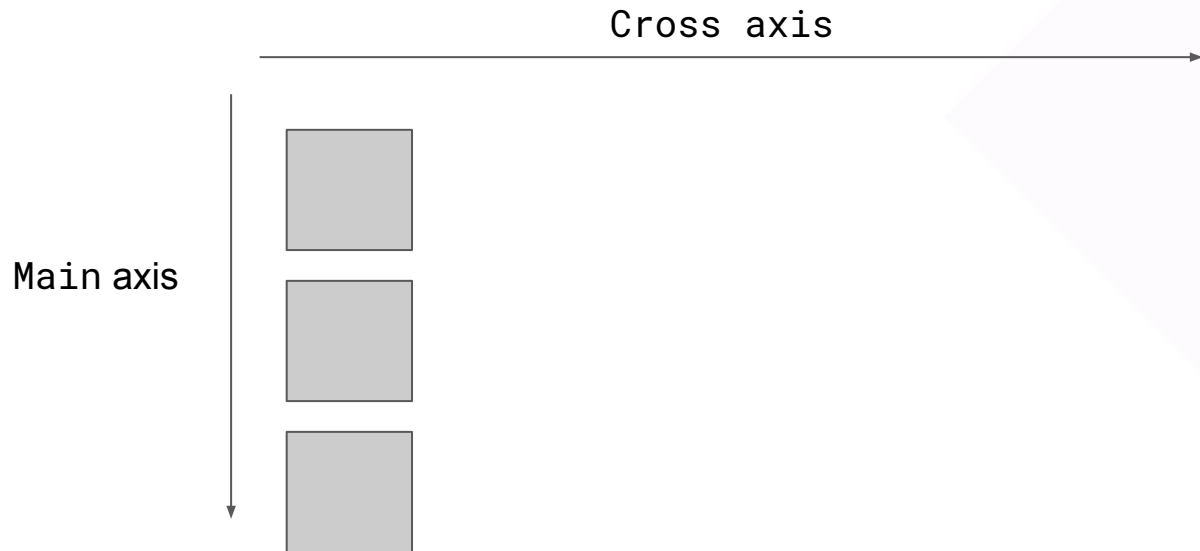
Main axis & Cross axis

flex-direction: row



Main axis & Cross axis

flex-direction: column



Exempel

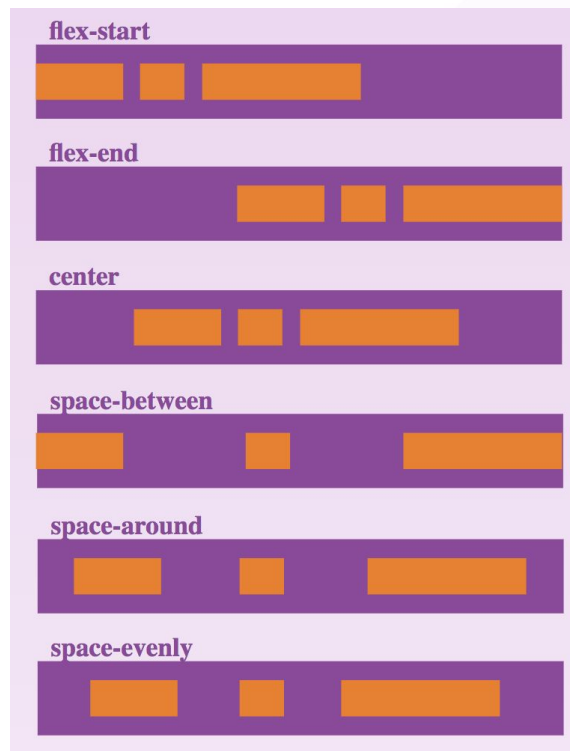
Flex-direction: Exempel

Flex alignments - main axis

`justify-content`

Styr positionering av flex-items på main-axeln.

[Bildkälla: CSS Tricks](#)

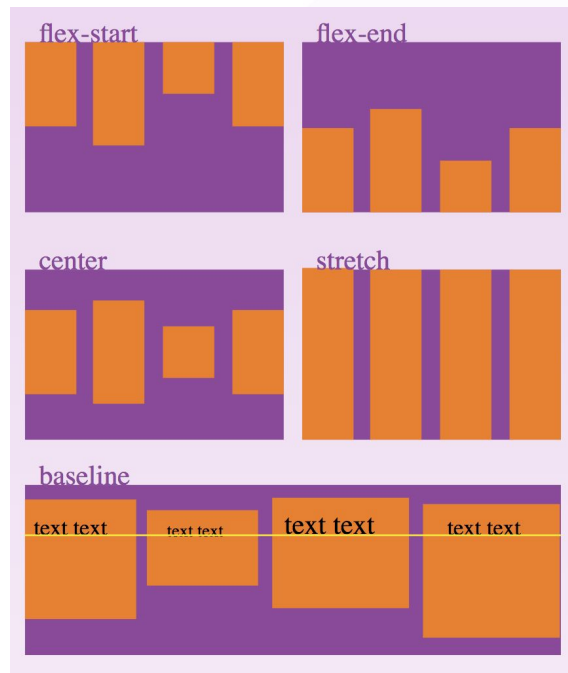


Flex alignments - cross axis

align-items

Styr positionering av flex-items på cross-axeln.

[Bildkälla: CSS Tricks](#)



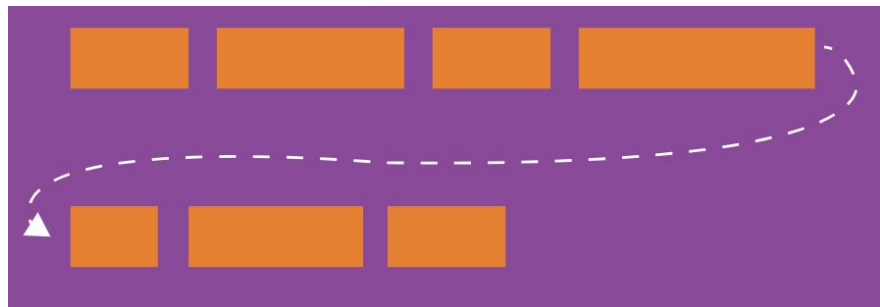
Exempel

Flex alignments: [Exempel](#)

Flex-wrap

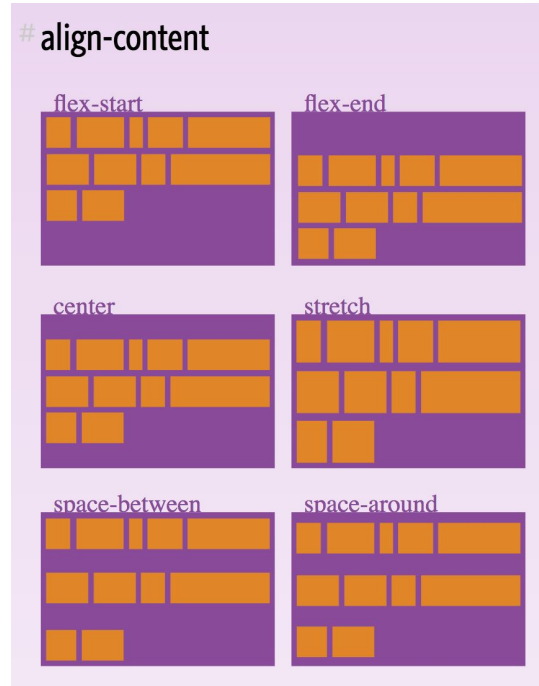
Visar flex-items på flera rader

- wrap
- wrap-reverse



Align-content

När `flex-wrap` används kan man använda `align-content` för att ändra position längs cross-axeln.

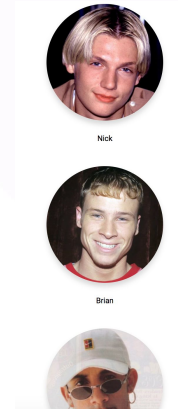


Exempel

Flex-wrap: [Exempel](#)

Flexbox - Backstreet Boys

Precis som vi gjorde med bootstrap så ska vi ska skapa fansida för Backstreet Boys i tre olika storlekar:



Hur börjar vi?

Vi börjar med att skapa en grundstruktur:

```
.member {  
  text-align: center;  
  margin: 10px;  
}  
  
.name {  
  margin: 20px;  
}
```

```
<div class="page">  
  <div class="member">  
    <div class="name">Nick</div>  
  </div>  
  <div class="member">  
    <div class="name">Brian</div>  
  </div>  
  <div class="member">  
    <div class="name">A.J.</div>  
  </div>  
  <div class="member">  
    <div class="name">Kevin</div>  
  </div>  
  <div class="member">  
    <div class="name">Howard</div>  
  </div>  
</div>
```

Resultat

display: flex

För att våra element ska bli flex-items använder vi oss av `display: flex`.

Vi vill också att våra element ska radbrytas på mindre skärmar, så vi använder `flex-wrap`.

```
.page {  
  display: flex;  
  flex-flow: wrap;  
  justify-content: center;  
}
```

[Resultat](#)

Exempel

Vi har ni en responsiv layout, men mönstret ser inte riktigt ut som på bilden ännu. Där vill vi ha 2 kolumner längst upp och 3 kolumner längst ner på de största skärmarna.



Nick



Brian



A.J.



Kevin



Howard

Rader

Vi skapar därför två olika rader för att dela in strukturen som på bilden.

```
<div class="page">
  <div class="page-row">
    <div class="member">
      <div class="name">Nick</div>
    </div>
    <div class="member">
      <div class="name">Brian</div>
    </div>
  </div>
  <div class="page-row">
    <div class="member">
      <div class="name">A.J.</div>
    </div>
    <div class="member">
      <div class="name">Kevin</div>
    </div>
  </div>
</div>
```

Rader

Vi ser nu att våra två element med `.page-row` har blivit våra flex-items istället.

Nick

A.J.

Brian

Kevin

Howard



Nick

Brian

A.J.

Kevin

Howard

Resultat

Flex i flex

För att göra våra .member-element till flex-items sätter vi display: flex på .page-row istället:

```
.page-row {  
  display: flex;  
  flex-flow: wrap;  
  justify-content: center;  
}
```

```
<div class="page">  
  <div class="page-row">  
    <div class="member">  
      <div class="name">Nick</div>  
    </div>  
    <div class="member">  
      <div class="name">Brian</div>  
    </div>  
  </div>  
  <div class="page-row">  
    <div class="member">  
      <div class="name">A.J.</div>  
    </div>  
    <div class="member">  
      <div class="name">Kevin</div>  
    </div>  
  </div>  
</div>
```

Resultat

Lägga in bilderna

Vi har nu rätt struktur och kan börja lägga in bilderna:

```
<div class="page">
  <div class="page-row">
    <div class="member">
      
      <div class="name">Nick</div>
    </div>
    <div class="member">
      
      <div class="name">Brian</div>
    </div>
  </div>
  <div class="page-row">
    <div class="member">
      
      <div class="name">A.J.</div>
    </div>
  </div>
</div>
```

```
img {
  width: 100%;
}
```

Resultat

Storleken

För att få rätt storlek använder vi `flex-basis`:

```
.member {  
  text-align: center;  
  margin: 10px;  
  flex-basis: 200px;  
}
```

[Resultat](#)

Flex-basis

Nu får vi äntligen rätt struktur på vårt innehåll.

Vi noterar att bilderna inte växer sig större än 200px (eftersom flex-grow är 0 default) men att de kan krympa när utrymmet blir för lite (eftersom flex-shrink är 1 default).

Vill vi vara övertydliga kan vi sätta de värdena med `flex: 0 1 200px` istället, men det kommer ge samma sak och det finns därför inte något större värde i att göra det.

Styling

Slutligen stylar vi bilderna och får följande resultat:

Resultat

Källor

- [CSS-tricks: A Complete Guide to Flexbox](#)
- [MDN: CSS Flexible Box Layout](#)

Övningar

<https://github.com/LinkNorth/HTMLCSS/tree/master/exercises/week6/repetition>