# Flexbox

Mer läsning om flexbox:

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

<https://scotch.io/tutorials/a-visual-guide-to-css3-flexbox-properties>

<http://www.sketchingwithcss.com/samplechapter/cheatsheet.html>

Med Flexbox deklarerar man först ett element som en flexbox-container genom att ange display: flex. Elementen som befinner sig i flex-containern lyder sedan under "flex-regler" vilket betyder:

* Att man får tillgång till flex-grow.
* Att blockelement fungerar annorlunda:
  + De lägger sig bredvid varandra istället för under
  + De expanderar så att de fyller hela höjden av det omkringliggande elementet. Istället för att bara vara stora nog att ha plats med sitt innehåll.
  + De krymper horisontellt så att de bara är stora nog att ha plats med sitt innehåll, istället för att växa till bredden hos det omkringliggande elementet.



Det grå blocket ovan är det element som fått display: flex.

## 

## 

## Flex-grow och flex-shrink

Hur mycket ett block i en flexbox-container ska växa:

flex-grow: 1;



Det högra elementet har fått flex-grow: 1 och därmed läggs det "tomma utrymmet" till elementets bredd.

När man använder flex och flex-grow händer det ibland att element "krymper" när flex-systemet försöker skapa utrymme åt elementen som får växa. Det kan man förhindra genom att lägga till följande CSS-regel till elementen som inte ska få krympa:

flex-shrink: 0;

## 

## 

## Flex-direction

Ibland vill man inte att elementen i en flex-container ska ligga bredvid varandra, utan mer likna den vanliga strukturen där block lägger sig under varandra, växer ut åt sidan och får begränsad höjd. Men man vill fortfarande ha kvar flex-grow. Då kan man lägga till flex-direction: column på flex-containern.

flex-direction: column;



I exemplet ovan har det grå blocket i den högra versionen av layouten fått flex-direction: column.

## 

## 

## Strategi för att dela upp en sida i lådor

Det här är en användbar strategi för att avgöra hur en layout ska delas upp.

Börja med en skiss av din valda layout:



Leta sedan rätt på de linjer du kan hitta som går **från kant till kant**.



I det här fallet finns det bara en sådan. Då vet vi att det yttersta elementet (body) kommer att behöva delas in i **två delar** och att riktningen ska vara **vertikal** (flex-direction: column). Vi kan också notera att den övre delen (header) nog ska ha en fast höjd, medan den nedre delen ska växa och bli lika hög som själva sidan. Steg 1 kommer alltså att vara att lägga in två element i body; en header och till exempel en div som ges en klass.

* body kommer att behöva display: flex så att vi får tillgång till flex-grow.
* body kommer också att behöva ha flex-direction: column.
* header kommer att behöva en fast höjd och antagligen flex-shrink: 0.
* diven kommer att behöva en flex-grow: 1.

Därefter gör vi samma undersökning av den övre och den nedre delen. Den övre, headern, ska inte delas in ytterligare. Där finns inga streck alls.

I den nedre delen däremot, finns ett lodrätt streck.



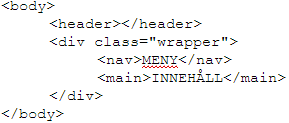
Det betyder att det i den nedre delen, diven, kommer det att ligga två element - nav och main.

* diven behöver display: flex, både för att se till så att elementen inuti diven lägger sig bredvid varandra och för att ge tillgång till flex-grow.
* nav ska troligen ha en fast bredd och flex-shrink: 0.
* main ska växa ut och uppta resten av bredden, och behöver därför flex-grow.

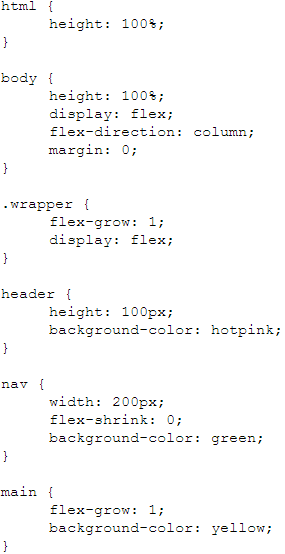
## 

## Exempelkod

HTML:



CSS:



Resultat:

