

Informatietechnologie 1

Cascading Style Sheets

Flexbox

Kristof Michiels

In deze presentatie

- Flexbox?
- Flexbox containers en items
- Flex-richting en flexen over meerdere lijnen/kolommen
- Flex elementen uitlijnen
- Elementen laten groeien en krimpen met "flex"

Flexbox?

Flexbox

- Module heet voluit CSS Flexible box layout module
- Zorgt voor controle over het plaatsen van elementen op één as (horizontaal of verticaal)
- Bijzonder geschikt voor menubalken en andere navigatie, produktlijstjes, afbeeldingsgalerijen, ...
- Zorgt er samen met Grid Layout (werkt op 2 assen) voor dat we fantastische layouts kunnen maken met CSS

Flexbox

- Elementen kunnen uitrekken of krimpen binnen hun containers
- Zo verhinderen we teveel of te weinig ruimte: super voor responsive web design
- Andere mogelijkheden:
 - Alle elementen zelfde hoogte te geven
 - Eenvoudig horizontaal of verticaal centreren
 - Volgorde van de elementen te veranderen (los van HTML-volgorde)

Flexbox containers en items

Een flexbox container aanmaken

- Flexbox is - net als inline, block en grid - een layout mode
- Een flexbox container aanmaken doe je dus met *display:flex*
- Op de volgende slide staat een basisvoorbeeld:
 - Standaard worden de divs als block elementen onder elkaar geplaatst
 - Als we de flexbox aanzetten gaan ze automatisch horizontaal in één rij staan
 - De elementen worden standaard in een rij van links naar rechts geplaatst

Een flexbox container

HTML

CSS

Result

EDIT ON
CODEPEN

LIVE

```
1 <div class="een">1</div>
2 <div class="twee">2</div>
3 <div class="drie">3</div>
4 <div class="vier">4</div>
5 <div class="vijf">5</div>
```

LIVE

```
1 :root {font-size: 62.5%;}
2
3 div {
4     height: 5rem;
5     font-size: 3rem;
6     color: white;
7     text-align: center;
8     font-family: helvetica;
9     box-sizing: border-box;
10    padding: 1rem;}
11
12 .een {background-color: #8A4F7D;}
13 .twee {background-color: #887880;}
14 .drie {background-color: #88A096;}
15 .vier {background-color: #BBAB8B;}
16 .vijf {background-color: #EF8275;}
17
18 body {
19     display: flex;
20     border: 0.1rem solid grey;}
21
```

Resources

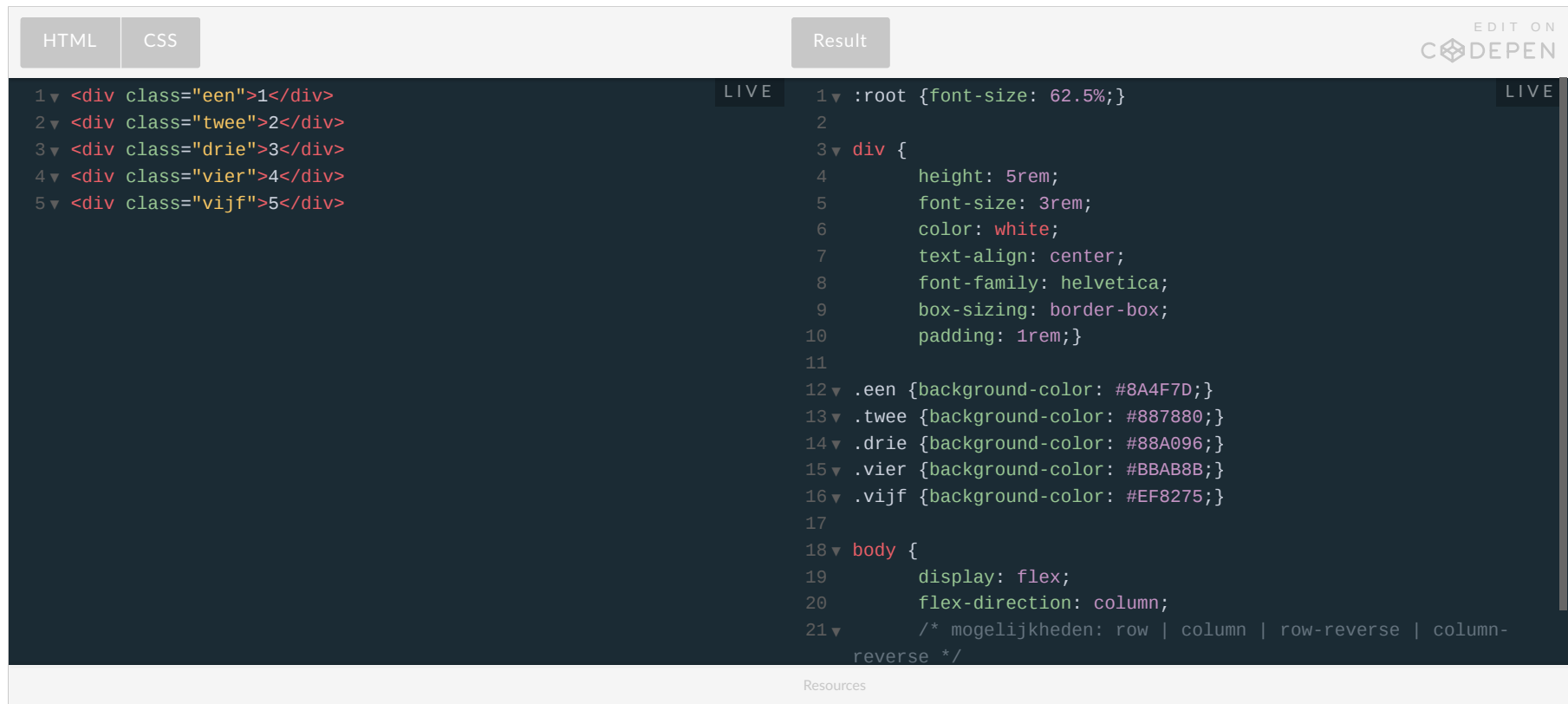
<https://codepen.io/admkrm/pen/KbgvLQ>

Flex-richting en flexen over meerdere lijnen/kolommen

De "flow" controleren met *flex-direction*

- Je kan binnen een flex-container controleren op welke manier de binnenste elementen geplaatst worden
- De opties die we hier hebben worden aangestuurd door de *flex-direction* eigenschap
- De waarden die we kunnen gebruiken: row (standaard) | column | row-reverse | column-reverse
- Bij *row* is de as horizontaal, bij *column* is de as verticaal

De "flow" controleren met *flex-direction*



The screenshot shows a CodePen editor with two tabs: 'HTML' and 'CSS'. The 'HTML' tab is active, displaying five div elements with classes 'een', 'twee', 'drie', 'vier', and 'vijf'. The 'CSS' tab is also visible, showing a root selector for font-size and a 'div' selector for height, font-size, color, text-align, font-family, box-sizing, and padding. Below these, there are five class selectors (.een, .twee, .drie, .vier, .vijf) each with a background-color. The 'body' selector is also present, setting display: flex and flex-direction: column. A comment in the CSS code lists possible flex-direction values: row, column, row-reverse, and column-reverse. The 'Result' tab is visible on the right, showing a 'LIVE' indicator. The CodePen logo and 'EDIT ON CODEPEN' text are in the top right corner.

```
HTML
1 <div class="een">1</div>
2 <div class="twee">2</div>
3 <div class="drie">3</div>
4 <div class="vier">4</div>
5 <div class="vijf">5</div>

CSS
1 :root {font-size: 62.5%;}
2
3 div {
4     height: 5rem;
5     font-size: 3rem;
6     color: white;
7     text-align: center;
8     font-family: helvetica;
9     box-sizing: border-box;
10    padding: 1rem;}
11
12 .een {background-color: #8A4F7D;}
13 .twee {background-color: #887880;}
14 .drie {background-color: #88A096;}
15 .vier {background-color: #BBAB8B;}
16 .vijf {background-color: #EF8275;}
17
18 body {
19     display: flex;
20     flex-direction: column;
21     /* mogelijkheden: row | column | row-reverse | column-
reverse */
```

<https://codepen.io/admkrm/pen/YdGrjV>

Uitsmeren over meerdere lijnen met *flex-wrap*

- Met *flex-wrap* bepaal je of de flexbox zich over meerdere lijnen mag uitstrekken
- De waarden die we kunnen gebruiken: nowrap (standaard) | wrap | wrap-reverse
- Standaard wordt alles op één lijn gezet

Uitsmeren over meerdere lijnen met *flex-wrap*

HTML

CSS

Result

EDIT ON CODEPEN

```
1▼ :root {font-size: 62.5%;}
2
3▼ div {
4    height: 5rem;
5    font-size: 3rem;
6    color: white;
7    text-align: center;
8    font-family: helvetica;
9    box-sizing: border-box;
10   padding: 1rem;}
11
12▼ body {
13   display: flex;
14   flex-direction: row;
15▼  /* mogelijkheden: row | column | row-reverse | column-
    reverse */
16   flex-wrap: wrap;
17▼  /* mogelijkheden: nowrap | wrap | wrap-reverse */
18   border: 0.1rem solid grey;}
19
20▼ .een {background-color: #8A4F7D;}
21▼ .twee {background-color: #887880;}
```

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

Resources

1×

0.5×

0.25×

Rerun

<https://codepen.io/admkrm/pen/LMROYy>

Uitsmeren over meerdere kolommen met *flex-wrap*

HTML

CSS

Result

EDIT ON
CODEPEN

1▼ :root {font-size: 62.5%;}

2

3▼ div {

4 height: 5rem;

5 font-size: 3rem;

6 color: white;

7 text-align: center;

8 font-family: helvetica;

9 box-sizing: border-box;

10 padding: 1rem;}

11

12▼ body {

13 display: flex;

14 flex-direction: column;

15 height: 35rem;

16▼ /* mogelijkheden: row | column | row-reverse | column-

reverse */

17 flex-wrap: wrap;

18▼ /* mogelijkheden: nowrap | wrap | wrap-reverse */

19 border: 0.1rem solid grey;}

20

21▼ .een {background-color: #8A4F7D;}

LIVE

1	8	15
2	9	16
3	10	17
4	11	18
5	12	19
6	13	20
7	14	

Resources

1× 0.5× 0.25×

Rerun

<https://codepen.io/admkrm/pen/JwROKN>

Flex-direction en flex-wrap verkort...

- Doe je met flex-flow
- Waarden: flex-direction flex-wrap

```
div.mijndiv {  
  display: flex;  
  height: 35rem;  
  flex-flow: column wrap;  
}
```

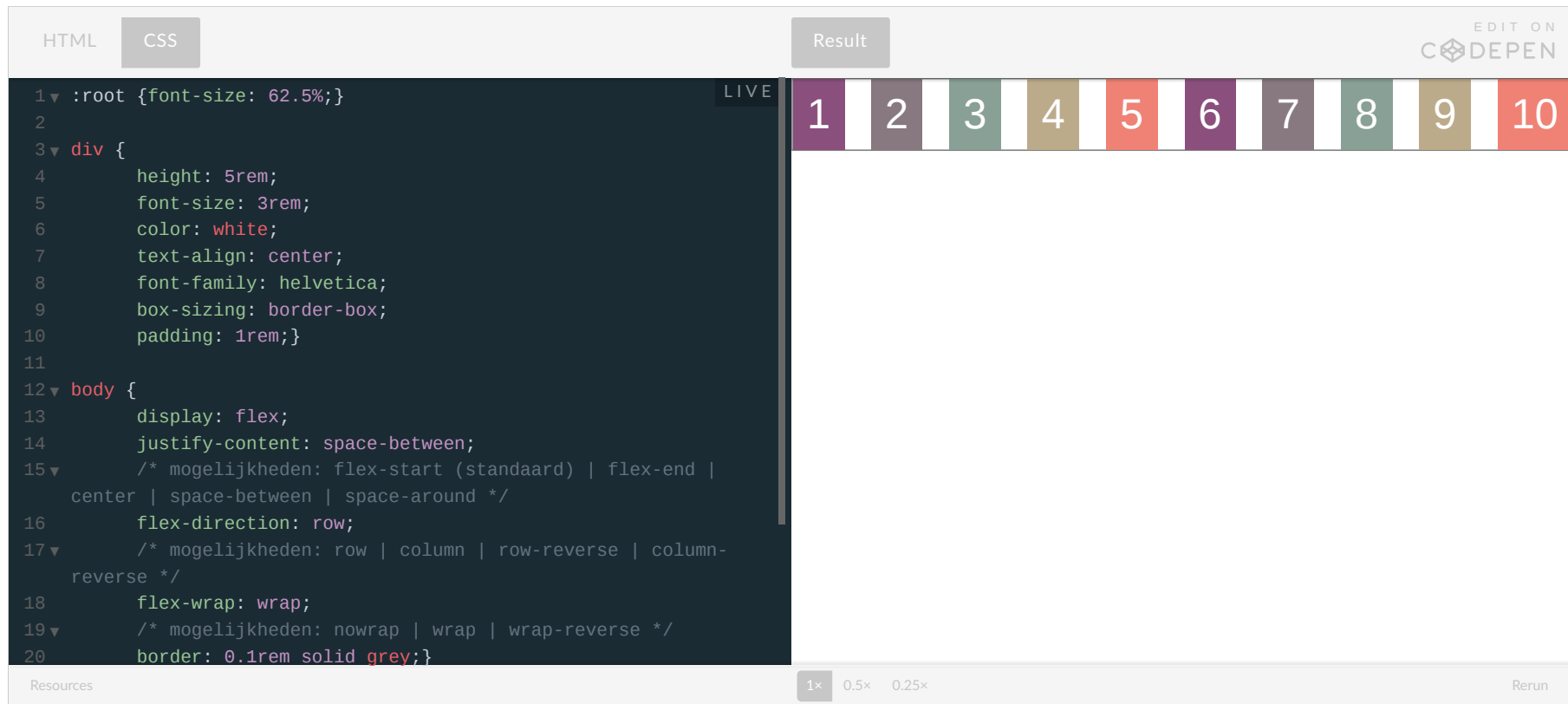
Flex elementen uitlijnen

Uitlijnen van elementen in de flexbox container

- Uitlijnen op de hoofdas doen we met *justify-content*
- Je past het toe op de flex-container
- Waarden: flex-start (standaard) | flex-end | center | space-between | space-around

```
div.mijndiv {  
  display: flex;  
  justify-content: flex-start;  
}
```

Uitlijnen van elementen in de flexbox container



The screenshot shows a CodePen live editor with the following CSS code:

```
1 :root {font-size: 62.5%;}
2
3 div {
4   height: 5rem;
5   font-size: 3rem;
6   color: white;
7   text-align: center;
8   font-family: helvetica;
9   box-sizing: border-box;
10  padding: 1rem;}
11
12 body {
13   display: flex;
14   justify-content: space-between;
15   /* mogelijkheden: flex-start (standaard) | flex-end |
16      center | space-between | space-around */
17   flex-direction: row;
18   /* mogelijkheden: row | column | row-reverse | column-
19      reverse */
20   flex-wrap: wrap;
21   /* mogelijkheden: nowrap | wrap | wrap-reverse */
22   border: 0.1rem solid grey;}
```

The result shows a horizontal row of 10 numbered items (1-10) with a light grey border. The items are evenly spaced, demonstrating the effect of `justify-content: space-between;` and `flex-direction: row;`.

<https://codepen.io/admkrm/pen/ebdeXW>

Uitlijnen van elementen (kolom)

HTML

CSS

Result

EDIT ON
CODEPEN

```
1▼ :root {font-size: 62.5%;}
2
3▼ div {
4    height: 5rem;
5    font-size: 3rem;
6    color: white;
7    text-align: center;
8    font-family: helvetica;
9    box-sizing: border-box;
10   padding: 1rem;}
11
12▼ body {
13   display: flex;
14   justify-content: space-between;
15▼  /* mogelijkheden: flex-start (standaard) | flex-end |
    center | space-between | space-around */
16   flex-direction: column;
17   height: 150vh;
18▼  /* mogelijkheden: row | column | row-reverse | column-
    reverse */
19   flex-wrap: wrap;
20▼  /* mogelijkheden: nowrap | wrap | wrap-reverse */
```

1

2

3

4

5

6

7

Resources

1x 0.5x 0.25x

Rerun

<https://codepen.io/admkrm/pen/ebdeXW>

Uitlijnen op de kruis-as

- Hiermee bedoelen we in de hoogte (bij row) en links/rechts (bij column)
- Uitlijnen op de kruis-as doen we met *align-items*
- Je past het toe op de flex-container
- Waarden: stretch (standaard) | flex-start | flex-end | center | baseline

```
div.mijndiv {  
  display: flex;  
  align-items: flex-start;  
}
```

Uitlijnen op de kruis-as

HTML

CSS

Result

EDIT ON
CODEPEN

```
1▼ :root {font-size: 62.5%;}
2
3▼ div {
4     height: 5rem;
5     font-size: 3rem;
6     color: white;
7     text-align: center;
8     font-family: helvetica;
9     box-sizing: border-box;
10    padding: 1rem;}
11
12▼ body {
13    display: flex;
14    align-items: center;
15▼    /* mogelijkheden: stretch (standaard) | flex-start | flex-end | center | baseline */
16    justify-content: flex-start;
17▼    /* mogelijkheden: flex-start (standaard) | flex-end | center | space-between | space-around */
18    flex-direction: row;
19    height: 20vh;
20▼    /* mogelijkheden: row | column | row-reverse | column-reverse */
```

LIVE

1 2 3 4 5 6 7 8 9 10

Resources

1x 0.5x 0.25x

Rerun

Individuele elementen uitlijnen

- Doe je op de kruis-as met *align-self*
- Waarden: flex-start | flex-end | center | baseline | stretch (standaard)

```
div {  
  align-self: flex-start;  
}
```

Meerdere flexlijnen uitlijnen

- Doe je met align-content
- Je bepaalt hiermee hoe je meerdere flexlijnen uitspreid over de kruis-as
- Waarden: flex-start | flex-end | center | space-around | space-between | stretch (standaard)

```
body {  
  display: flex;  
  flex-direction: row;  
  height: 60vh;  
  align-content: center;  
  flex-wrap: wrap;  
  border: 0.1rem solid grey;}
```

Meerdere flexlijnen uitlijnen

HTML

CSS

Result

EDIT ON
CODEPEN

```
:root {font-size: 62.5%;}

div {
  height: 5rem;
  font-size: 3rem;
  color: white;
  text-align: center;
  font-family: helvetica;
  box-sizing: border-box;
  padding: 1rem;
  width: 25%;}

body {
  display: flex;

  flex-direction: row;
  height: 60vh;
  align-content: center;
  /* Waarden: flex-start | flex-end | center | space-around |
  space-between | stretch (standaard) */
  flex-wrap: wrap;
  border: 0.1rem solid grey;}

.een {background-color: #8A4F7D;}
```

1	2	3	4
5	6	7	8
9	10		

Resources1x0.5x0.25xRerun

<https://codepen.io/admkrm/pen/qLapov>

Flexbox op de navigatiebalk

HTML

CSS

Result

EDIT ON
CODEPEN

LIVE

```
1▼ ul {
2    display: flex;
3    align-items: center;
4    background-color: teal;
5    padding: .5em;
6    margin: 0;
7    color: white;
8    height: 3rem;
9 }
10
11▼ li {
12    margin: 0 1rem;
13 }
14
15▼ li.logo {
16    margin-right: auto;
17 }
18 }
```

Mijn logo

[Navigatie 1](#)

[Navigatie 1](#)

[Navigatie 1](#)

Resources

1× 0.5× 0.25×

Rerun

<https://codepen.io/admkrm/pen/BvLJeW>

Elementen laten groeien en krimpen met "flex"

Bepalen hoe elementen "flexen" in de container

- We kunnen hiermee automatisch elementen laten groeien en krimpen
- Doen we met de flex-eigenschap
- Deze eigenschap wordt op de elementen gezet, niet op de container (kan individueel)
- Waarden: none | 'flex-grow flex-shrink flex-basis'
- Standaard: 0 1 auto;
- 1 staat voor "zet aan", 0 voor "zet uit"
- De flex-basis: zet de startafmeting. Je kan dit ook gebruiken om de width te zetten

"Flexen" met flex

HTML

CSS

Result

EDIT ON CODEPEN

```
1▼ :root {font-size: 62.5%;}
2
3▼ div {
4     font-size: 3rem;
5     color: white;
6     text-align: center;
7     font-family: helvetica;
8     box-sizing: border-box;
9     padding: 1rem;
10    flex: 1 1 auto;}
11
12▼ body {
13    display: flex;
14    align-items: stretch;
15▼    /* mogelijkheden: stretch (standaard) | flex-start | flex-
    end | center | baseline */
16    flex-direction: row;
17    height: 30vh;
18    border: 0.1rem solid grey;}
19
20▼ .een {background-color: #8A4F7D;}
21▼ .twee {background-color: #887880;}
```

1

2

3

4

5

6

7

8

9

10

Resources

1× 0.5× 0.25×

Rerun

<https://codepen.io/admkrm/pen/vvXdmy>

Spelen met flex en flex-grow

HTML

CSS

Result

EDIT ON
CODEPEN

```
1▼ :root {font-size: 62.5%;}
2
3▼ div {
4     font-size: 3rem;
5     color: white;
6     text-align: center;
7     font-family: helvetica;
8     box-sizing: border-box;
9     padding: 1rem;
10    flex: 1 1 auto;}
11
12▼ body {
13    display: flex;
14    align-items: stretch;
15    flex-direction: row;
16    height: 30vh;
17    border: 0.1rem solid grey;}
18
19▼ .vier {
20    flex: 5 1 auto;}
21
22▼ .een {background-color: #8A4F7D;}
```

1

2

3

4

5

6

7

8

9

10

Resources

1x 0.5x 0.25x

Rerun

<https://codepen.io/admkrm/pen/REGQQV>

Shortcuts voor de flex eigenschap

- *flex: initial*: zelfde als `flex: 0 1 auto`; Elementen zullen niet automatisch groeien, maar wel krimpen
- *flex: auto*: zelfde als `flex: 1 1 auto`; Elementen zullen groeien en krimpen
- *flex: none*: zelfde als `flex: 0 0 auto`; Niet krimpen en niet groeien
- *flex: getal*: zelfde als `flex: getal 1 0px`; Absolute flex. Getal bepaalt hoeveel vrije ruimte het element krijgt

De volgorde van flex elementen aanpassen

- Soms handig bij responsive web design
- Vooral nuttig indien je flexbox gebruikt voor layout en wij gaan dat zelden doen
- Je gebruikt hiervoor de eigenschap order
- Werkt op zelfde manier als z-index bij positioning
- Elk element heeft standaard 0. Je kiest getal zelf. Negatief getal kan ook

De volgorde aanpassen met order

HTML

CSS

Result

EDIT ON
CODEPEN

```
1▼ div {
2    font-size: 3rem;
3    color: white;
4    text-align: center;
5    font-family: helvetica;
6    box-sizing: border-box;
7    padding: 1rem;
8    flex: 1 1 auto;}
9
10▼ body {
11    display: flex;
12    align-items: stretch;
13    flex-direction: row;
14    height: 30vh;
15    border: 0.1rem solid grey;}
16
17▼ .een {order: 2;}
18▼ .twee {order: 1;}
19▼ .drie {order: 3;}
20▼ .vier {order: 5;}
21▼ .vijf {order: 4;}
22
```

212354

Resources1x0.5x0.25xRerun

<https://codepen.io/admkrm/pen/VgKQoz>

IT1 - Flexbox

Vragen?

kristof.michiels01@ap.be