

# Information Technology 3

## Flexbox

**Kristof Michiels**

# In this presentation

- Flexbox?
- Flexbox containers and items
- Flex-direction\_ and flexing over multiple lines/columns
- Aligning flexbox elements
- Letting elements grow and shrink with "flex"

**Flexbox?**

# Flexbox

- Full name of the CSS module: CSS Flexible box layout module
- Gives control for layouting elements on 1 AXIS (horizontal or vertical)
- We may use it for menus and other navigation, lists of items, image galleries, ...
- Grid Layout (works in 2 dimensions) remains primary tool for page layout, supported in some cases by Flexbox
- Together they form a fantastic duo for building extremely solid layouts

# Flexbox

- Elements can grow or shrink within their containers
- Super voor responsive web design
- Other possibilities:
  - Give all elements same heights
  - centering elements horizontally or vertically
  - Changing the order of elements without changing the HTML

# Flexbox containers and items

# Making a flexbox container

- Flexbox is - like inline, block and grid - a layout mode
- So you create a flexbox container with *display:flex*
- On the next slide you'll find a basic example:
  - If we turn on the flexbox container the 5 divs are placed horizontally in a row
  - The 5 elements are placed automatically from left to right, taking no more than their natural width

# A 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>

8 / 32



## Flex-direction and flexing over multiple lines/columns

# Controlling "flow" with flex-direction

- Flex-direction allows you to control how the elements are placed in the container
- Values we can use: row (standard) | column | row-reverse | column-reverse
- Use row and the axis is horizontal, use column and it is vertical

# Controlling "flow" with flex-direction

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     flex-direction: column;
21     /* mogelijkheden: row | column | row-reverse | column-
reverse */
```

Resources

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

# Flexing over multiple lines with flex-wrap

- With flex-wrap you can allow the flexbox container to spread over multiple lines
- Values: nowrap (standard) | wrap | wrap-reverse

# Flexing over multiple lines with 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>

# Flexing over multiple lines with 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>

# Aligning flex elements

# Aligning flex elements

- Aligning on the head-axis is done with justify-content
- It is applied on the flex container
- Values: flex-start (standard) | flex-end | center | space-between | space-around

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



# Aligning flex elements (horizontal axis)

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: row;
17▼  /* mogelijkheden: row | column | row-reverse | column-
    reverse */
18   flex-wrap: wrap;
19▼  /* mogelijkheden: nowrap | wrap | wrap-reverse */
20   border: 0.1rem solid grey;}
```

1 2 3 4 5 6 7 8 9 10

1x 0.5x 0.25x Rerun

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

# Aligning flex elements (vertical axis)

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/roMYXz>

# Aligning on the cross-axis

- By which we mean the height (with flex-direction:row) and left/right (with flex-direction:column)
- You do it using the align-items property
- You place align-items on the flex container
- Values: stretch (standard) | flex-start | flex-end | center | baseline

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

# Aligning on the cross-axis

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 */
```

1 2 3 4 5 6 7 8 9 10

1x 0.5x 0.25x Rerun

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

# Aligning individual elements(on cross-axis)

- You do it with align-self
- Values: stretch (standard) | flex-start | flex-end | center | baseline

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

# Aligning multiple flex-lines

- You do it with align-content
- Determines how you spread multiple flex-lines over the cross-axis
- Values: stretch (standard) | flex-start | flex-end | center | space-around | space-between

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

# Aligning multiple flex-lines

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   width: 25%;}
12
13▼ body {
14   display: flex;
15
16   flex-direction: row;
17   height: 60vh;
18   align-content: center;
19▼  /* Waarden: flex-start | flex-end | center | space-around
   | space-between | stretch (standaard) */
20   flex-wrap: wrap;
21   border: 0.1rem solid grey;}
```

1	2	3	4
5	6	7	8
9	10		

Resources1x0.5x0.25xRerun

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

# Flexbox in a navigation menu

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▼ li {  
10   margin: 0 1rem;}  
11▼ li.logo {  
12   margin-right: auto;}
```

My logo

[Navigation 1](#)

[Navigation 2](#)

[Navigation 3](#)

Resources

1× 0.5× 0.25×

Rerun

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



Letting elements grow and shrink with "flex"

# Letting elements grow and shrink with "flex"

- With the flex property we can make elements grow and shrink automatically
- This property is placed on the elements, not the flex container
- Values: none | 'flex-grow flex-shrink flex-basis'
- Default: 0 1 auto;
- 1 means "turn on", 0 voor "turn off"
- Flex-basis: sets the starting width. You may use it to set the width of your element

# "Flexing" with 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>

# Toying with flex and 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 for the flex property

- *flex:initial*: same as flex: 0 1 auto;
- *flex: auto*: same as flex: 1 1 auto;
- *flex: none*: same as flex: 0 0 auto;
- *flex: number*: same as flex: number 1 0px; =Absolute flex. Number decides how much room the element gets

# Changing the order of flex elements

- With the property `order`
- Good for responsive web design
- Especially if you use flexbox for layouting and we won't :-)
- Works the same way as z-index with positioning
- Standard value for each element is 0

# Changing the order

HTML

CSS

Result

EDIT ON  
CODEPEN

LIVE

```
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
```

2

1

3

5

4

Resources

1× 0.5× 0.25×

Rerun

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

IT1 - Flexbox

Questions?

[kristof.michiels01@ap.be](mailto:kristof.michiels01@ap.be)