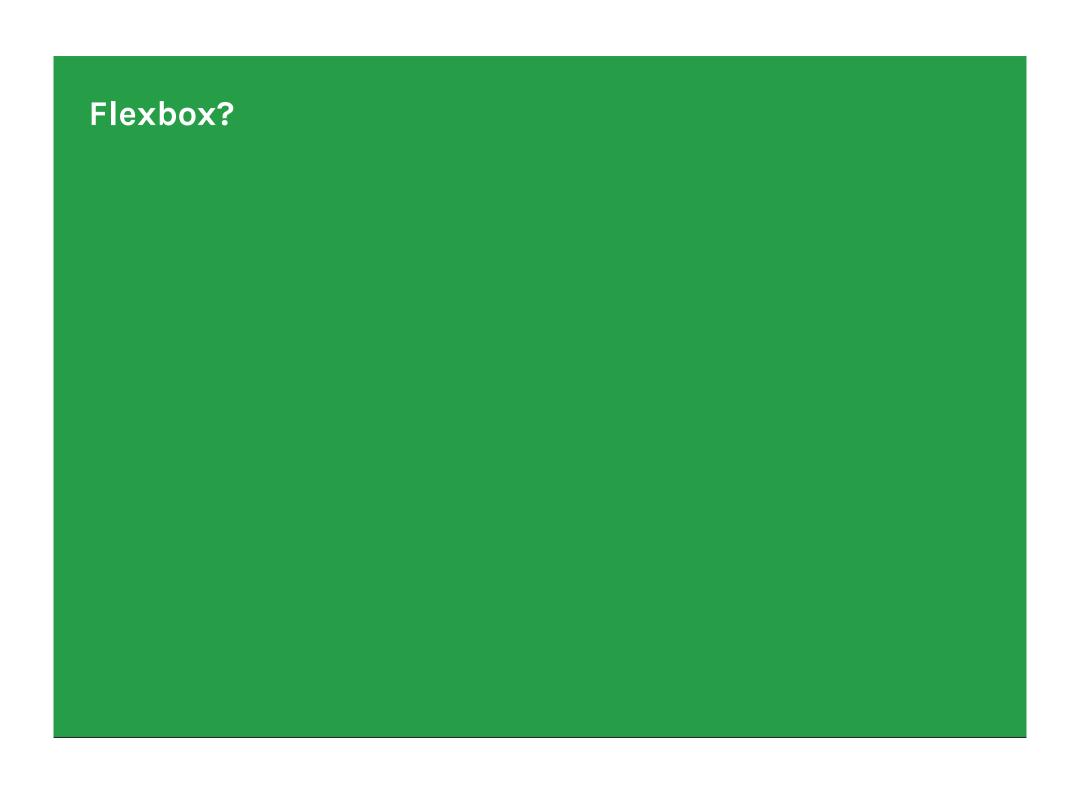
Information Technology 3

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

- 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



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

```
CODEPEN
                                                               LIVE
                                                                        1 v :root {font-size: 62.5%;}
1 v <div class="een">1</div>
2 v <div class="twee">2</div>
3 √ <div class="drie">3</div>
4 v <div class="vier">4</div>
                                                                                 height: 5rem;
5 v <div class="vijf">5</div>
                                                                                 font-size: 3rem;
                                                                                 text-align: center;
                                                                                 box-sizing: border-box;
                                                                                 padding: 1rem;}
                                                                       12 v .een {background-color: #8A4F7D;}
                                                                       13 ▼ .twee {background-color: #887880;}
                                                                       14 ▼ .drie {background-color: #88A096;}
                                                                       15 v .vier {background-color: #BBAB8B;}
                                                                       16 v .vijf {background-color: #EF8275;}
                                                                                 border: 0.1rem solid grey;}
```

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

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

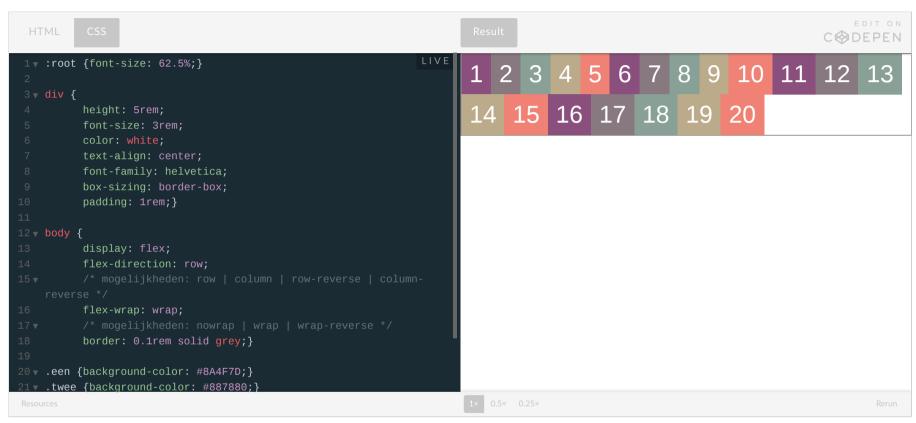
```
CODEPEN
                                                               LIVE
                                                                        1 v :root {font-size: 62.5%;}
                                                                                                                                        LIVE
1 ▼ <div class="een">1</div>
2 v <div class="twee">2</div>
3 v <div class="drie">3</div>
4 v <div class="vier">4</div>
                                                                                 height: 5rem;
5 v <div class="vijf">5</div>
                                                                                 font-size: 3rem;
                                                                                 text-align: center;
                                                                                 box-sizing: border-box;
                                                                                 padding: 1rem;}
                                                                       12 v .een {background-color: #8A4F7D;}
                                                                       13 v .twee {background-color: #887880;}
                                                                       14 v .drie {background-color: #88A096;}
                                                                       15 v .vier {background-color: #BBAB8B;}
                                                                       16 v .vijf {background-color: #EF8275;}
                                                                                 flex-direction: column;
```

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



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

Flexing over multiple lines with flex-wrap



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



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

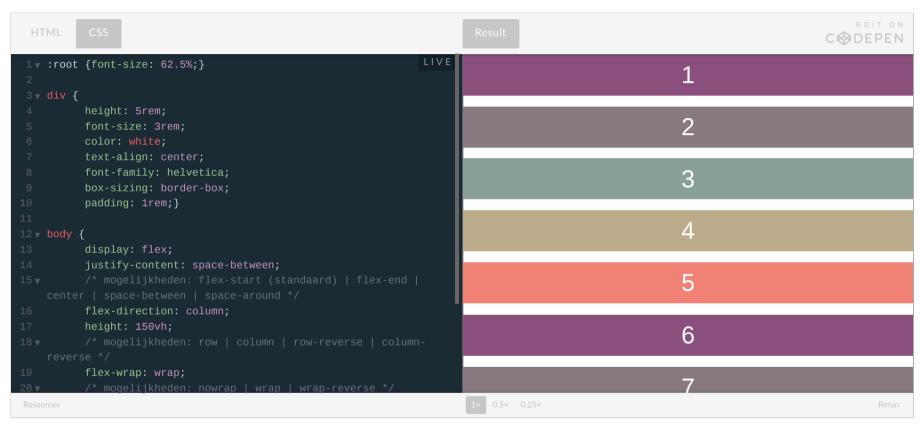
```
div.mydiv {
    display: flex;
    justify-content: flex-start;
}
```

Aligning flex elements (horizontal axis)



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

Aligning flex elements (vertical axis)



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



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

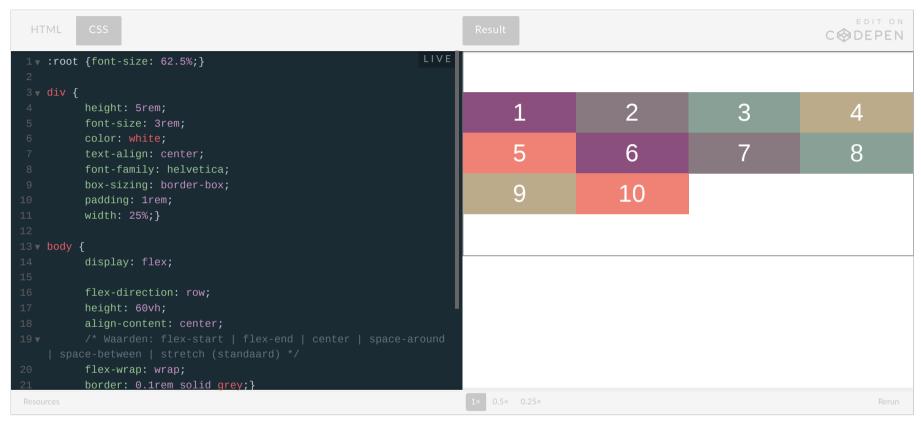
```
div {
    align-self: flex-start;
}
```

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

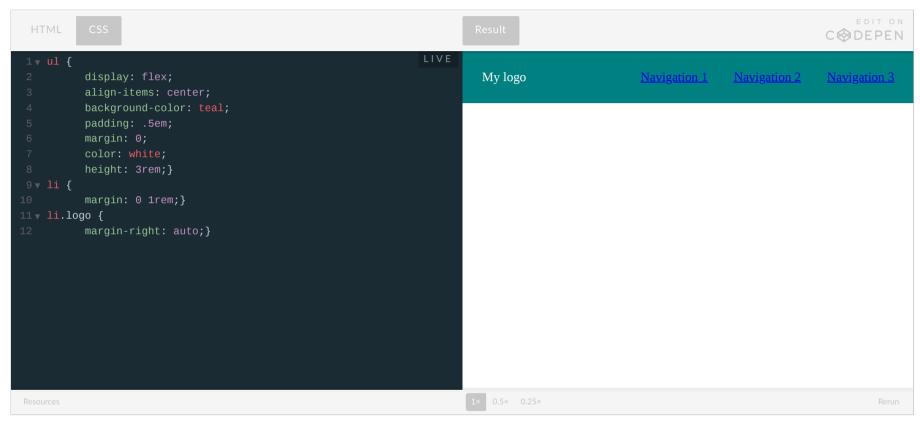
```
body {
    display: flex;
    flex-direction: row;
    height: 60vh;
    align-content: center;
    flex-wrap: wrap;
    border: 0.1rem solid grey;}
```

Aligning multiple flex-lines



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

Flexbox in a navigation menu



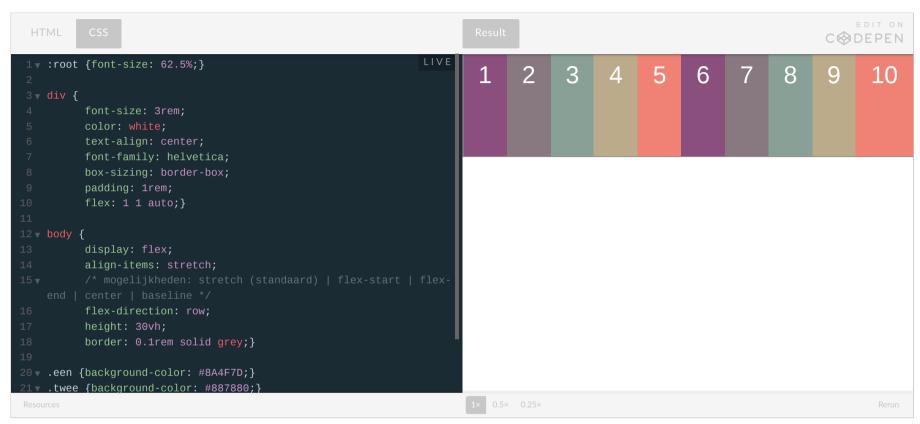
https://codepen.io/admkrm/pen/BvLJeW



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'
- Dafault: 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



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

Toying with flex and flex-grow



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 border
- Good for responsive web design
- Expecially 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



https://codepen.io/admkrm/pen/VqKQoz

IT1 - Flexbox

Questions?

kristof.michiels01@ap.be