

# CSS – part 2

ILKKA KYLMÄNIEMI

# CSS Layout

- ▶ Layout = arrange elements (boxes) in relation to the viewport and to one another
- ▶ With CSS you control where elements are positioned relative to the following factors:
  - ▶ default position in normal layout flow
  - ▶ the other elements around them
  - ▶ their parent container,
  - ▶ the main viewport/window

# Layout techniques

- ▶ Normal flow
- ▶ The display property
- ▶ Flexbox
- ▶ Grid
- ▶ Floats
- ▶ Positioning
- ▶ Table layout
- ▶ Multiple-column layout



# Normal flow

- ▶ the way that Block and Inline elements are displayed on a page before any changes are made to their layout.
- ▶ inline elements display in the inline direction = in the direction words are displayed in a sentence
- ▶ block elements display one after the other, as paragraphs do

# The display property

- ▶ allows us to change the default way something displays
- ▶ Example:
  - ▶ `<li>` element is `display: block` by default => list items display one below the other
  - ▶ change the display value to `inline` => list items display next to each other

```
li {  
  display: inline;  
}
```

- ▶ Hiding elements is usually done with `display: none;`

# Flexbox + Grid

- ▶ Flexbox = Flexible Box Layout

- ▶ [MDN Link](#)

- ▶ Grid Layout

- ▶ [MDN Link](#)

# Example

- ▶ Open CSS3 folder from HTML + CSS Examples in IDE/editor

# Floats + Positioning

- ▶ Float

- ▶ Before flexbox, floating was used for page layouts, but you should not use it for that anymore.
- ▶ [MDN Link](#)

- ▶ Positioning

- ▶ allows you to move an element from where it would otherwise be placed in normal flow over to another location
- ▶ [MDN Link](#)
- ▶ When using absolute, fixed and sticky in some element, the parent element should be positioned with relative

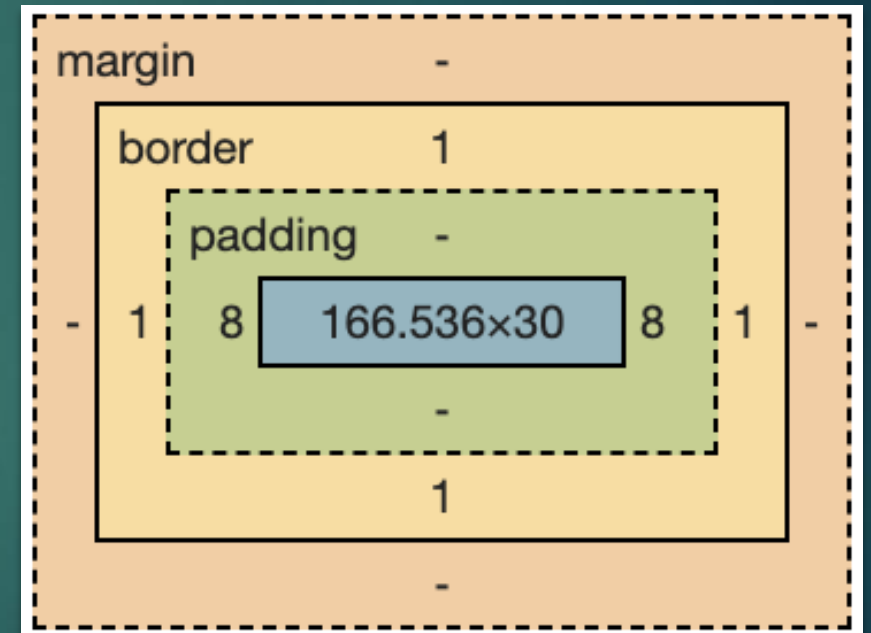


# Table layout + Multi-column layout

- ▶ Many decades ago, before CSS, this technique was used to do layouts for the whole page. Don't do it anymore even if some random Google result might tell you to do so.
- ▶ Multi-column layout
  - ▶ Again, do not use to do full page layout, just some part of the layout e.g. `<article>`
  - ▶ [MDN Link](#)

# THE BOX MODEL

- ▶ All HTML elements can be considered as boxes.
- ▶ Box model is basically a box that wraps around every HTML element.
- ▶ Box model consists of:
  - ▶ Margin
  - ▶ Border
  - ▶ Padding
  - ▶ Content



# Margin, border, padding

- ▶ Margin
  - ▶ the space around the outside of the border.
- ▶ Border
  - ▶ the solid line that is just outside the padding.
- ▶ Padding
  - ▶ the space around the content.

# Box sizing

- ▶ Standard
  - ▶ Any padding and border is added to the width/height of the box to get the total size taken up by the box
  - ▶ This is the default behaviour
- ▶ Alternative
  - ▶ No need to add up the border and padding to get the real size of the box.
  - ▶ In CSS:
    - ▶ `box-sizing: border-box;`

# Media Queries

- ▶ key part of responsive web design
- ▶ allow you to create different layouts depending on the size of the viewport
- ▶ can also be used to detect other things about the environment/browser like touch screen or mouse (hover abilities)
- ▶ Mobile first:

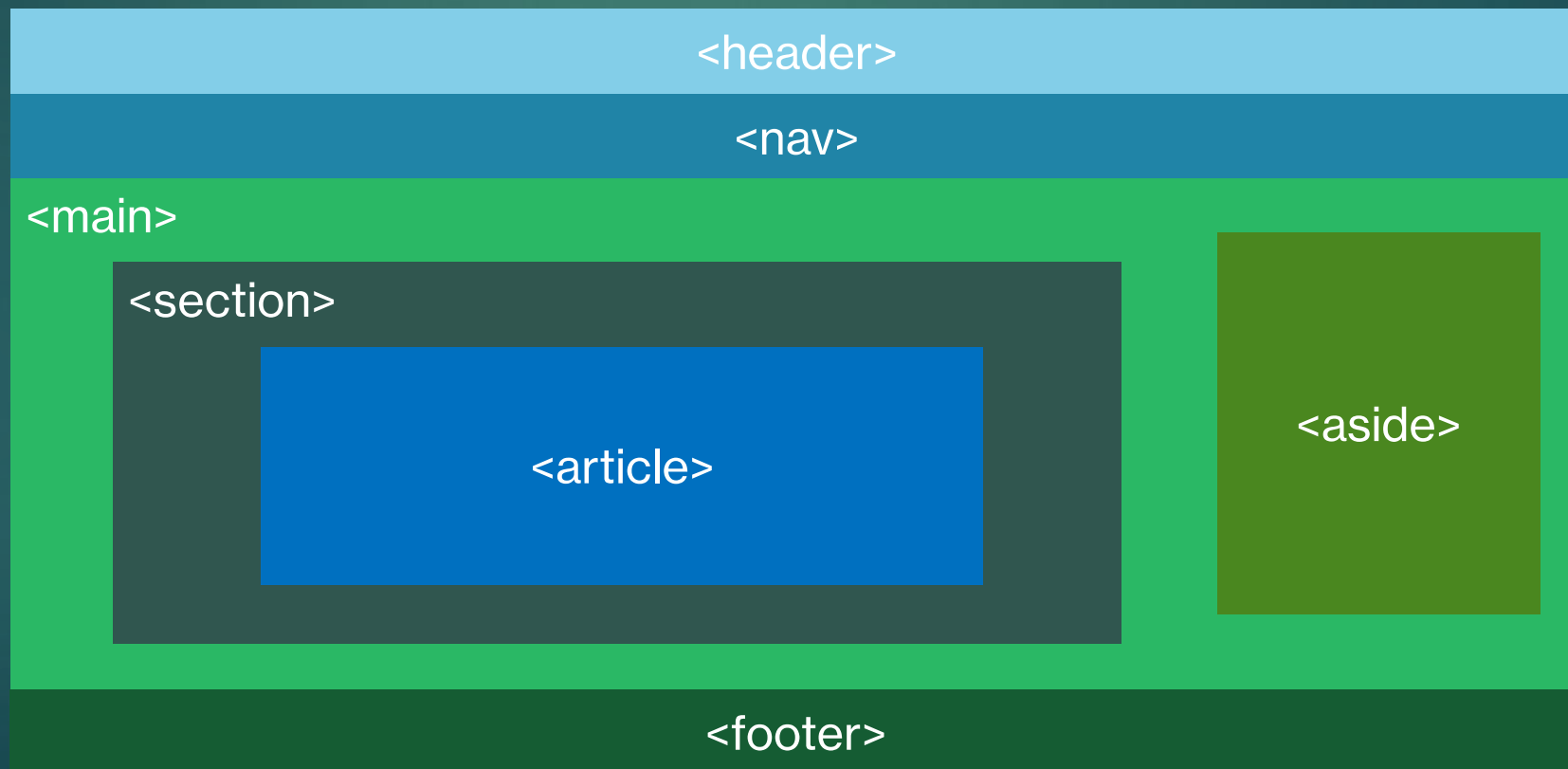
```
/* rules for small screen */  
.container {  
  width: 100%;  
}  
  
/* rules for medium and bigger screens */  
@media screen and (min-width: 769px) {  
  .container {  
    width: 80%;  
  }  
}
```

- ▶ [MDN Link](#)



# Example

- ▶ Open CSS4 folder from HTML + CSS Examples in IDE/editor



# Links

- ▶ [Complete guide to flexbox](#)
- ▶ [Viewport meta tag](#)
- ▶ [Figure](#)
- ▶ [Vector graphics](#)