Cascading Style Sheets

GERARD ROMA 2022

Browser technologies

Behaviour

Text, structure

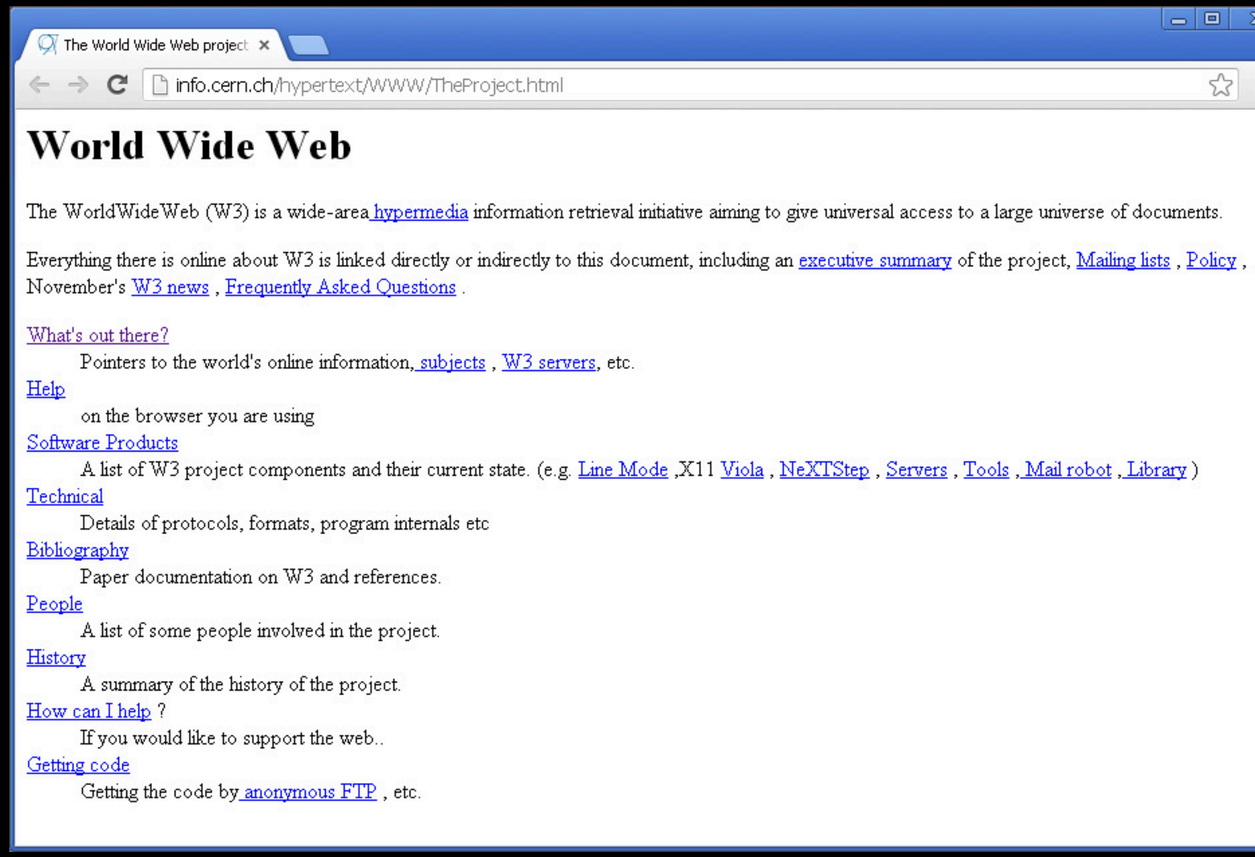
Appearance

Before the Web

- Books, newspapers and magazines had consistent styles for different functions:
  - Regular text
  - Headings
  - Highlights
  - Pictures
  - Captions
  - Page numbers
  - Column divisions
  - …

Before CSS

- The web was about sharing documents (initially scientific).
- HTML had basic formatting capabilities



HTML head

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>HTML Head Example</title>
    <link rel="stylesheet" href="local/style.css">
    <link rel="stylesheet" href="http://server.com/remote/style.css">
    <script src="local/index.js"></script>
    <script src="http://server.com/remote/index.js"></script>
  </head>
  <body>
  </body>
</html>
```

Cascading Style Sheets (CSS)

- Declarative language for describing the appearance of an HTML document.
- Usually written in separate css files.
- Can also be found inside <style> tags or inline style.

Best

```
<link rel="stylesheet" href="style.css">
```

Not ideal

```
<style>
p{
    font-size:16px
}
</style>
```

Avoid

```
<p style = "font-size:16px"></p>
```

Syntax

- A CSS is organised by rules.
- Each rule specifies:
  - What HTML elements it applies to
  - A list of values for different style properties

```
selector {
 property:value;
 property:value;
}
```

```
h1 {
 color: blue;
 font-size:25px;
}
```

Selectors

- All elements of this type
- Multiple elements
- Nested element

```
h1 {
 color: blue;
 font-size:25px;
}

h1, h2, h3 {
 font-family:Arial, sans-serif;
}

ul li {
  list-style-type:none;
}
```

Select by id

HTML

```
<div id="container">
contents
</div>
```

CSS

```
#container{
  border: 1px solid blue;
  padding:10px;
  background-color:grey;
}
```

Select by class

HTML

```
<p class="post">
some text
</p>
```

CSS

```
.post{
  color:white;
  background-color: black;
}
```

Combinations

HTML

```
<p id="opening">
  I am going to
<span class=“highlighted">
highlight
</span>
 some text
</p>
```

CSS

```
body p#opening
span.highlighted {
  color: red;
}
```

Select everything:

```
* {
  color: red;
}
```

Cascading, specificity, inheritance

- Cascading: rules from different sources may apply
- If two rules target the same element, the most specific takes precedence.
- If two rules are equally specific, the last one is applied.
- Some properties (e.g. colour) are inherited by child elements. Some are not (e.g. width, height).
- Cascading rules can be overridden with !important keyword.

Fonts

- By default, fonts can only be used if they are installed in the client computer.
- Alternatively, they can be downloaded (e.g. google fonts): add a link element to the html head.
- A small number of **web safe** fonts are available in all browsers.
- If the font is not found, the browser selects the closest one.

Font properties

```
p {
  font-family: Arial, Helvetica, sans-serif;
  font-style: italic;
  font-variant: small-caps;
  font-size: 15px;
  font-weight: bold;
}

body {
  font: italic small-caps bold 12px Georgia, serif;
}
```

Font sizes

- Font sizes can be determined by user preferences, so relative measures are usually preferable
- Several units:
  - keyword (medium, large, small…): very loose
  - px: size in pixels (precise, but may not hold)
  - pt: used in print media
  - %: relative to parent
  - em: relative to default font size

Colour properties

- color: foreground (typically text) color
- background-color: background of block or inline element
- border-color: colour of block or inline element border (can also be specified via **border** property

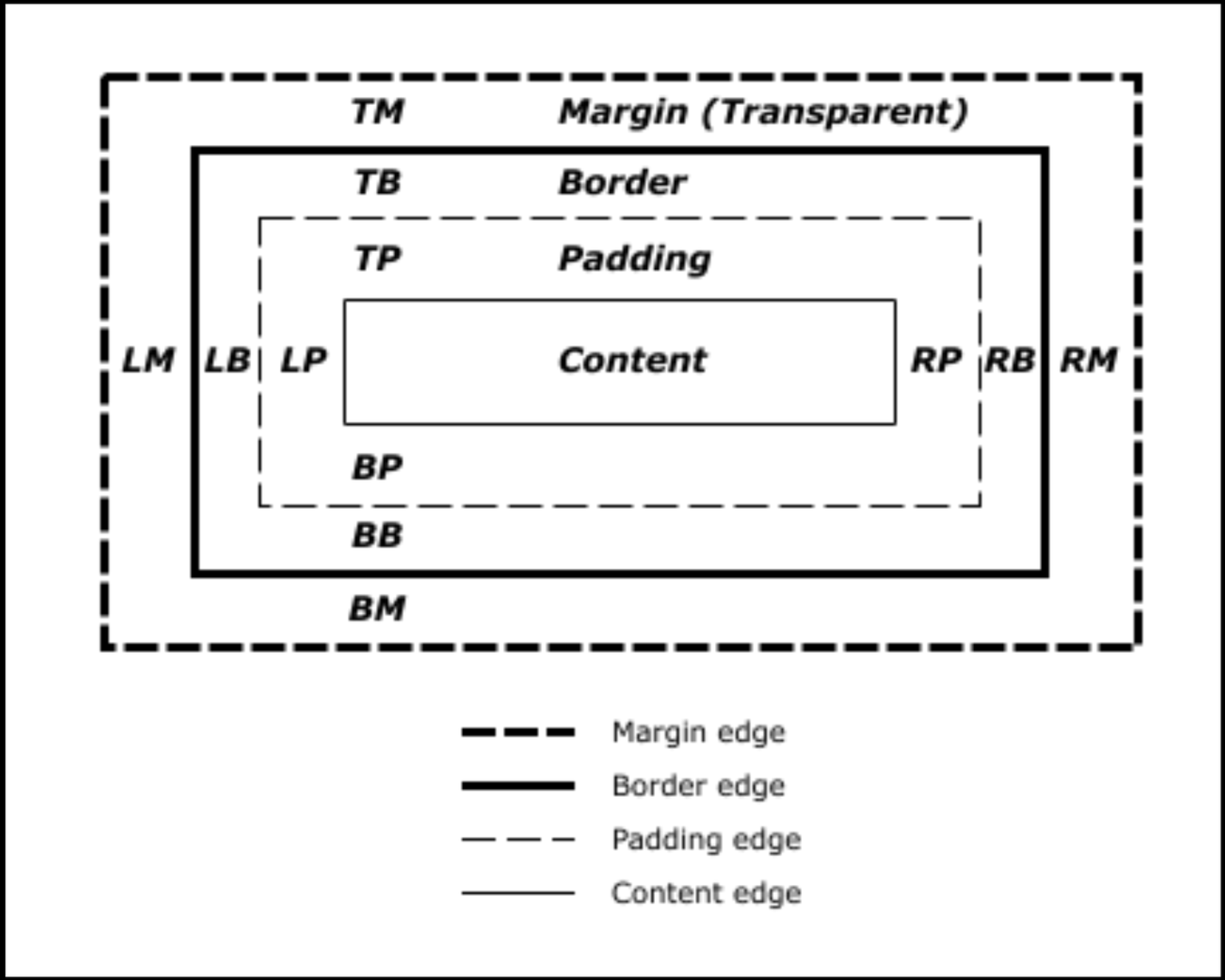
Colours

- Name
- Hex number
- RGB / RGBA
- HSL / HSLA

```
h1 {
  color: red;
}
h1 {
  color: #FF0000;
}
h1 {
  color: rgb(255, 0, 0);
}
h1 {
  color:hsla(0, 100%, 50%, 0.3)
}
```

Box model

- Every HTML element is seen as a box.
- Text and nested elements are the “Content”. The other properties control the spacing around it.



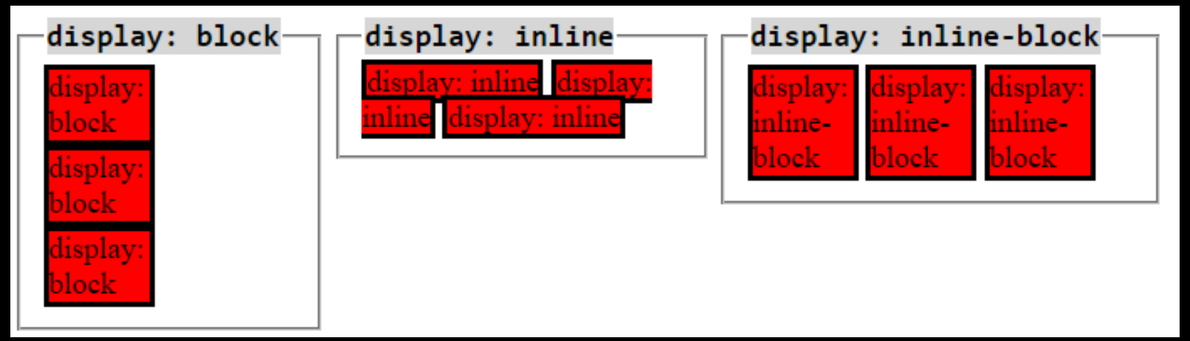
Box model properties

- Margin and padding are transparent.
- Border has thickness, colour and style.
- All properties can be specified for one side (left, right, top, bottom)
- Width and Height apply to content, width of margin, border and padding add to it.

```
div {
  width: 320px;
  padding: 10px;
  border: 5px solid gray;
  margin-left: 0;
  margin-right: 10px;
}
```

Display

- Display property can have many values, but most often:
  - Inline
  - Block
  - Inline-block
- Affects how box model properties are applied.
- Mainly: inline elements cannot have width and height, nor top or bottom margin / padding



Positioning

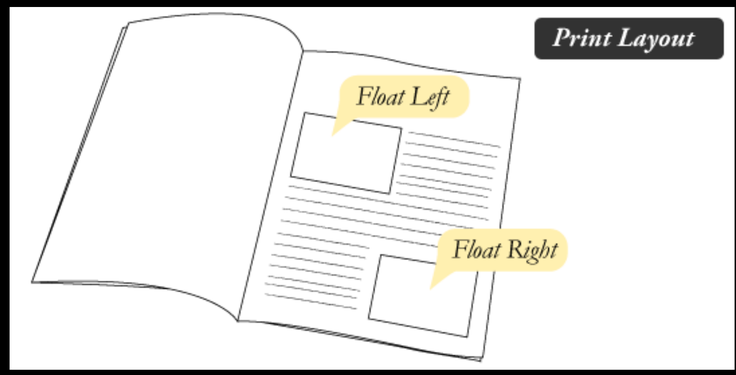
- Important: positioning of HTML elements is influenced by a number of factors:
  - Window size (resizable by the user)
  - Device
  - Font size (potentially modified by user)
  - CSS
- A common impulse is thinking in absolute terms, yet this is rarely possible or useful.

Position properties

- Position is controlled by: left, right, top, bottom.
- AND position property:
  - static - follows flow of page, ignore all position properties
  - relative - relative to where it should have been (!)
  - fixed - relative to viewport, regardless of scrolling
  - absolute - relative to the nearest ancestor that has the position property defined

Floats

- Float property allows text to flow around the target element
- Clear property: push element below floated elements (left, right or both)



Stack order

- Z-index: defines a virtual “z axis” (from screen to user)
- Controls what elements are visible when they overlap (e.g. using absolute positioning)
- Specified with an arbitrary number

```
img {
  position: absolute;
  left: 0px;
  top: 0px;
  z-index: -1;
}
```

Responsive design

- Web design is a form of graphic design, and shares many techniques with traditional print design.
- However, web sites are viewed in multiple devices, and are subject to user interaction with the format (browser size, fonts…)
- **Responsive design** is the principle of designing web pages so they will adapt to different devices and conditions.

Media queries

- Media queries select rules that only apply to specific devices or conditions.
- Can be specified in the <link> element or inside the stylesheet.
- The syntax involves a **media type** and a boolean expression involving **media features.**
- Media queries are an essential tool for **responsive design.**

Media queries syntax

Syntax

```
@media not|only mediatype and (mediafeature and|or|not mediafeature) {
  CSS-Code;
}
```

Examples

```
<link rel="stylesheet" media="screen and (min-width: 900px)" href="widescreen.css">
<link rel="stylesheet" media="screen and (max-width: 600px)" href="smallscreen.css">
```

```
@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```

(source: w3schools)