

CSS



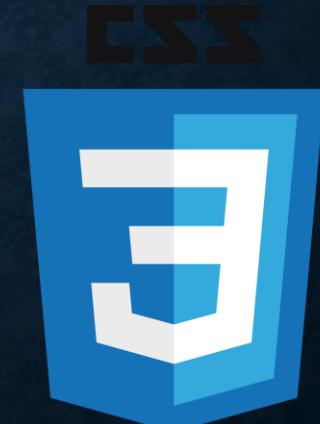
Syntax, integration, colors, Units, dimensions, and basics

COS216
AVINASH SINGH
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF PRETORIA

CSS – OVERVIEW

- Cascading Style Sheets (CSS)
- Language to visually style web pages
- Initial release in 1996
- Current version CSS3 (proposed in 1999, only adopted in ± 2014)
- Proposed version CSS4 (still in progress)

- MIME type: text/css
- Extension: .css



CSS – OVERVIEW

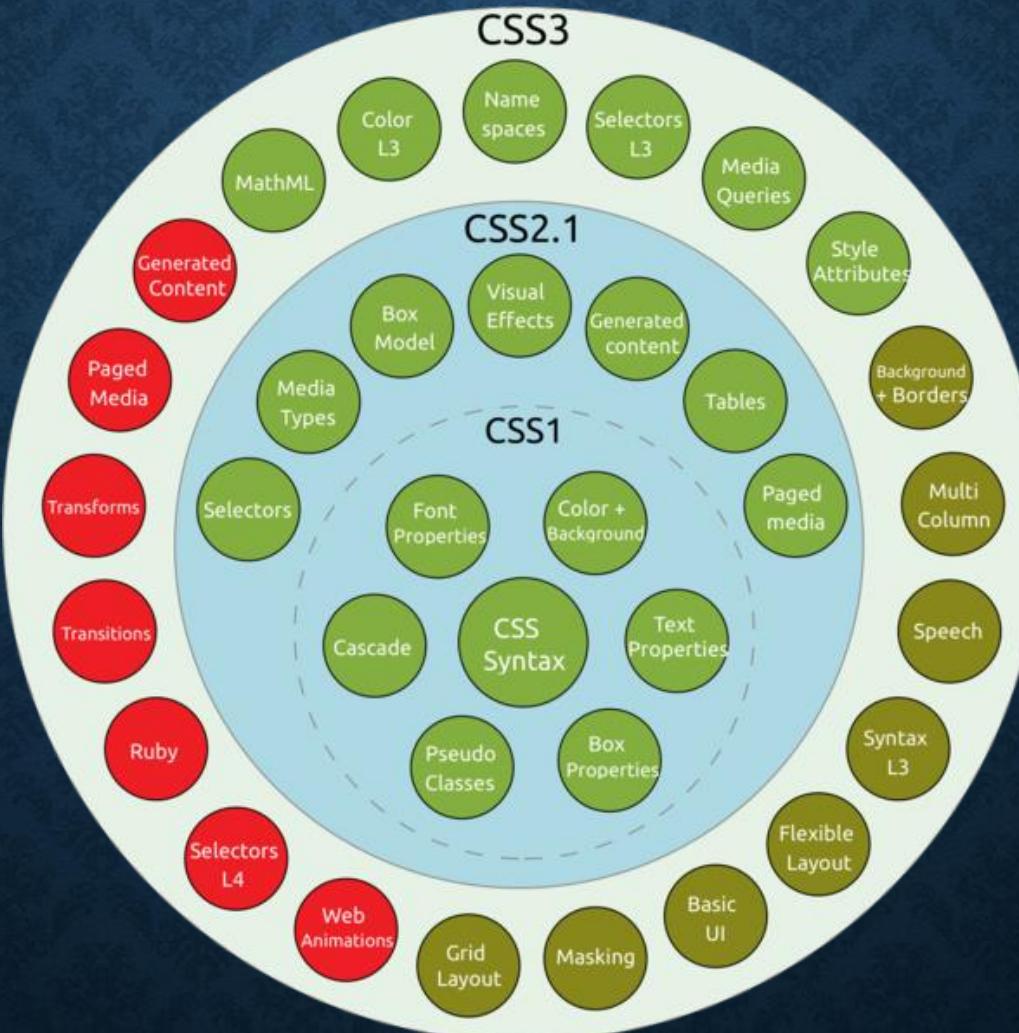
- Created by
 - Håkon Wium (Opera browser)
 - Lie Bert Bos (Argo Browser)
- Currently maintained by W3C



CSS – OVERVIEW

- CSS is a cascading language
 - Describes the priority with which rules should be applied
 - For each rule a cascade/priority/weight is assigned
 - Based on the cascade some rules are used or ignored, and others are overwritten
-
- For instance, an entire class of elements use a rule
 - Some specific elements of that class then overwrite certain rules
 - Example: common button style, specific style for small/medium/large buttons

CSS – OVERVIEW



CSS – BROWSERS

- There is an official CSS standard
- CSS3 took more than 15 years to be adopted
- Therefore, many browsers implement their own custom CSS attributes in the meantime, which often can only be rendered in that browser family
 - Mozilla
 - Chrome
 - Microsoft (IE & Edge)
 - Some others
- Hence, CSS is not always consistent across all browsers

CSS – WEBKIT

- Layout engine for web page rendering
- Created by Apple, Google, Adobe, KDE, and others
- Used by Safari, Chrome, Kindle, Nokia, Blackberry, various Unix browsers
- WebKit introduced some new JS and CSS functionality
- Many of their CSS additions were later incorporated into CSS3
- You might come across WebKit CSS selectors:

-webkit-<attribute name>



CSS – MOZILLA

- Similarly, Mozilla introduced their own CSS attributes while waiting for CSS3
- Mainly used for Firefox and Firefox related browsers
- You might come across Mozilla CSS selectors:

-moz-<attribute name>



CSS – MICROSOFT

- Similarly, Microsoft introduced their own CSS attributes while waiting for CSS3
- Used for Internet Explorer and Old Edge
- You might come across Microsoft CSS selectors:

-ms-<attribute name>



CSS – OPERA

- Similarly, Opera introduced their own CSS attributes while waiting for CSS3
- Used for the Opera browser
- You might come across Opera CSS selectors:

-o-<attribute name>



CSS – OTHER BROWSERS

- Some other browsers also had their own selectors
- With the adoption of CSS3, most these browser-specific selectors were replaced by the official CSS3 ones
- You can still provide a CSS rules with a bunch of browser-specific attributes
- Browsers will then pick the correct rule depending on their support
- Hence you can have one CSS style that has individual rule for standard CSS, WebKit, Mozilla, Microsoft, and Opera

CSS – SYNTAX

- Each CSS entity has a selector (specifies the element/group to style)
- Each CSS entity has 0 or more rules (name-value pairs)

```
selector
{
    name1: value1;
    name2: value2;
    name3: value3;
```

CSS – SELECTORS

- Selectors can be by
 - Tag
 - ID
 - Class
 - A combination of those
- More to follow in later lecture

CSS – COMMENTS

- Only multiline comments can be used in CSS (`/* ... */`)
- Single lines comments do not work (`// ...`)

```
/* Some comment here */
selector
{
    name1: value1;
    name2: value2;
    name3: value3; /* Some other comment */
}
```

CSS – INTEGRATION

- CSS can be integrated into HTML
 - Import an external CSS sheet
 - Provide CSS inside the HTML style tag
 - Provide inline CSS inside the HTML style attribute

CSS – INTEGRATION

- Import an external CSS sheet
- Note that most browser do **not** allow the link tag to be closed
 - <link></link> or <link .../>

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" type="text/css" href="mystyle.css">
    </head>
    <body>
        <div></div>
    </body>
</html>
```

CSS – INTEGRATION

- Provide CSS inside the HTML style tag

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div
      {
        width: 100px;
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

CSS – INTEGRATION

- Provide inline CSS inside the HTML style attribute

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <div style="width: 100px"></div>
  </body>
</html>
```

CSS – COLORS

- Colors can be represented as
 - As RGB (red-green-blue) octets using the **rgb()** or **rgba()** functions
 - As HSL (hue-saturation-lightness) octets using the **hsl()** or **hsla()** functions
 - Using RGB hexadecimal octets or nibbles (half octets)
 - Using standard web color names

CSS – COLORS

- CSS color function
- Each channel is represented as an octet (decimal integer between 0 and 255)
- Three channel (red, green, blue)

```
color: rgb(142, 14, 0);
```

- Four channel (red, green, blue, alpha)
- Alpha channel for transparency (decimal float between 0.0 and 1.0)

```
color: rgba(142, 14, 0, 0.5);
```

CSS – COLORS

- Hexadecimal (0 – 9 and A – F)
- Each octet is represented as 1 or 2 hexadecimal values
- 0 is black, F is white

```
color: #0F2;  
color: #0A3852;
```

- Most browsers do **not** support the alpha channel in hex colors
- There is not even a standard where the alpha channel (XX) should be placed

```
#XX0F21A5 /* Some graphic tools put alpha first */  
#0F21A5XX /* Some graphic tools put alpha last */
```

CSS – COLORS

- Convert between RGB and HEX

<https://www.webpagefx.com/web-design/hex-to-rgb/>

- Convert between RGB and HEX

<https://www.rapidtables.com/convert/color/rgb-to-hsl.html>

CSS – COLORS

- You can also use predefined color names
- About 150 different predefined colors

```
color: red;  
color: black;
```

- View the list of colors:

<https://www.webpagefx.com/web-design/hex-to-rgb/>

CSS – SIZES

- Resolutions (width and height) can be represented as
 - Percentage (%)
 - Pixels (**px**)
 - Metric/meters (**mm**, **cm**)
 - Imperial/inches (**in**)
 - Relative to font sizes of other elements (**em**, **ex**, **rem**)
 - Relative to viewport (**vw**, **vh**, **vmax**, **vmin**)
 - Relative to zero (**ch**)
- If no unit is provided, some browser will fail, others might default to pixel

CSS – SIZES

- If no unit is provided, some browser will fail, others might default to pixel
- Mostly percentage and pixels are used
- Percentages are used for dynamic resizing
- Pixels are used for fixed sizes
- Relative sizes are mainly used for fonts (especially font sizes on mobile phones)

```
width: 50%;  
width: 150px;  
width: 13cm;  
width: 2in;  
width: 30em;
```

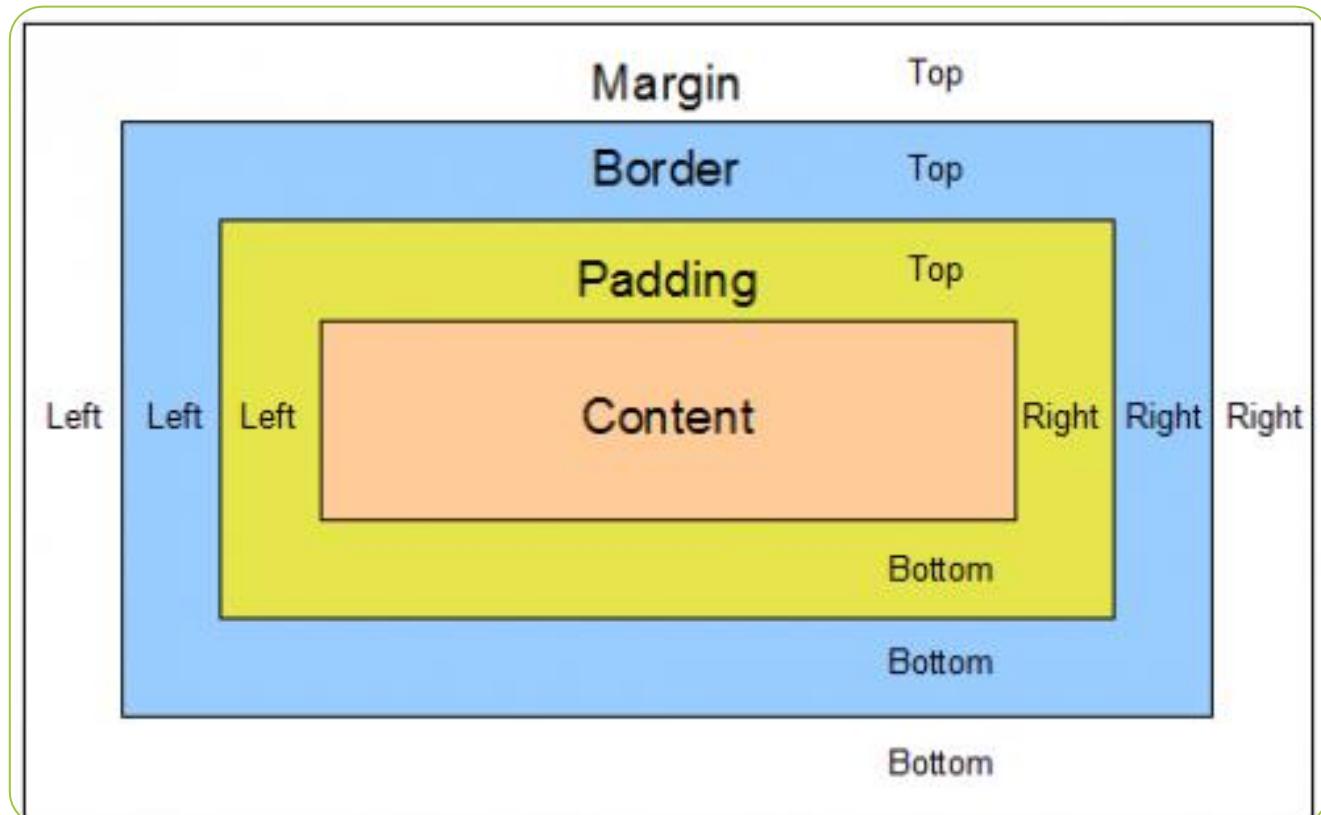
CSS – SIZES

- Convert between EM and PX

https://www.w3schools.com/cssref/css_pxtoemconversion.asp

CSS – DIMENSIONS

- Each HTML has various dimensions
 - Resolution – content (width, height)
 - Padding – inside element (top, bottom, left, right)
 - Border – around element (top, bottom, left, right)
 - Margin – outside element (top, bottom, left, right)



CSS – DIMENSIONS

- Each of these can be specified in various units (% , px, cm, in, etc)
- Margins are outside the element
 - Hence a 20px top margin will move the element down from the parent's top by 20px
- Padding is inside the element
 - Hence a 20px top padding will move the children inside the element down by 20px

CSS – WIDTH AND HEIGHT

- Different dimensions:
 - Current dimension (width, height)
 - Maximum allowed dimension (max-width, max-height)
 - Minimum allowed dimension (min-width, min-height)

CSS – WIDTH AND HEIGHT

```
div {  
    width: 100%; /* current 100% width of parent */  
    height: 512px; /* current 512px fixed width */  
  
    min-width: 100px; /* 100px minimum width */  
    min-height: 50px; /* 50px minimum height */  
  
    max-width: 2000px; /* 2000px maximum width */  
    max-height: 100%; /* 100% maximum height */  
}
```

CSS – MARGIN AND PADDING

- Each direction (top, bottom, left, right) can be
 - Specified for all
 - Specified individually
 - Specified as a group (top, bottom, left, right)
 - Specified as a group in main directions (top-bottom, left-right)

CSS – MARGIN

```
div
{
    margin: 5px 4px 3px 2px; /* 5px top, 4px right, 3px bottom, 2px left */
    margin: 5px 3px; /* 5px top and bottom, 3px left and right */
    margin: 5px; /* 5px top, left, bottom, right */

    margin-left: 5px; /* 5px left */
    margin-right: 5px; /* 5px right */
    margin-top: 5px; /* 5px top */
    margin-bottom: 5px; /* 5px bottom */
}
```

CSS – PADDING

```
div
{
    padding: 5px 4px 3px 2px; /* 5px top, 4px right, 3px bottom, 2px left */
    padding: 5px 3px; /* 5px top and bottom, 3px left and right */
    padding: 5px; /* 5px top, left, bottom, right */

    padding-left: 5px; /* 5px left */
    padding-right: 5px; /* 5px right */
    padding-top: 5px; /* 5px top */
    padding-bottom: 5px; /* 5px bottom */
}
```

CSS – BORDERS

- Borders have a
 - Color
 - Width
 - Style
 - Solid
 - Dotted
 - Dashed
 - Ridge
 - Double
 - Groove
 - None
 - More ...

CSS – BORDERS

```
div
{
    border: 2px solid red; /* can be specified in any order */
    border-color: green; /* set color only */
    border-width: 3px; /* set width only */
    border-style: dotted; /* set style only */
    border-bottom: 2px dotted lime; /* set specific direction */
    border-left-color: black; /* set specific direction and color */
}
```

CSS – RADIUS

- A border can have rounded corners, called a radius
- The radius cannot be specified as part of the border property
- The radius property has to be specified separately

CSS – RADIUS

```
div {  
    border-radius: 5px; /* all 4 corners rounded */  
    border-radius: 5px 3px 2px 4px; /* 4 corners with different radius */  
    border-radius: 5px 3px 2px; /* 3 corners with different radius */  
    border-radius: 5px 3px; /* top-right and left-bottom radius */  
    border-top-left-radius: 2px; /* specific corner radius */  
}
```

CSS – POSITION

- Displays an element at a specific position
- Fixed: Positioned relative to browser window (stays in place while scrolling)
- Absolute: Positioned to first non-static ancestor
- Relative: Positioned relative to normal position (immediate parent)
- Sticky and static: not that important
- Absolute and fixed can be used to disjoin elements from its parents and place them on top of other elements or the page

CSS – POSITION

```
div  
{  
    position: absolute;  
    position: relative;  
}
```

CSS – POSITION

- Once the position is specified, elements can be moved/offset in 4 directions
 - Top
 - Bottom
 - Left
 - Right
- Note that you will have to specify the position property together with the offset

CSS – POSITION

```
div
{
    /* top-right positioned in parent */
    position: relative;
    top: 10px;
    right: 30px;
}

div
{
    /* bottom-left positioned in window */
    position: fixed;
    left: 10px;
    bottom: 30px;
}
```

CSS – DISPLAY

- Defined the display behaviors of elements
- The browser will resize, position, and adjust the element accordingly
- **None**: do not display (hides the element)
- **Block**: display on separate line (standard behavior of a div)
- **Inline-block**: display on the same separate line (standard behavior of a span)
- **Table**: various display behaviors of tables (table, row, cell, caption, etc)
- **Flex**: various flexible/dynamic boxes (flex, inline-flex)

CSS – DISPLAY

```
div
{
    display: none; /* hide element */
    display: inline-table; /* table on the same line */
    display: block; /* block on a separate line */
    display: inline-flex; /* flex on same line */
}
```

CSS

