Styling with CSS

Documentation + tutorials:

- https://developer.mozilla.org/en-US/docs/Web/CSS
- https://www.w3schools.com/Css/

Works and projects making heavy use of CSS:

- Dina Kelberman, Untitled Game https://dinakelberman.com/untitledgame/
- Olia Lialina My boyfriend came back from the war

CSS - Cascading Style Sheets

HTML is not meant to style (inline styling eg: `<h1 style="color:red;background:black;">` is discouraged, however still sometimes useful to use).

CSS is the preferred to way to style.

- HTML tell the browser what content it should display
- CSS tells the browser **how to display** that content.

CSS inside an HTML page

You can do this in two ways:

- CSS code goes **inside the style tag** `<style> ... </style>`
- `<style> ... </style>` tags are **placed inside the head** of the HTML page

```
<!DOCTYPE html>
<html>
 <head>
    <style>

     body{
        background: #FF19DC;
            color: black;
            font-family: monospace;
         }

   </style>
   </head>
   </html>
   <body></body>
</html>
```

CSS in a separate stylesheet

The CSS for a HTML page (or several pages) can stored outside the page, in a **CSS file**.

To do that we need link the HTML file to the CSS file, using the `<link>` inside the `<head>`.

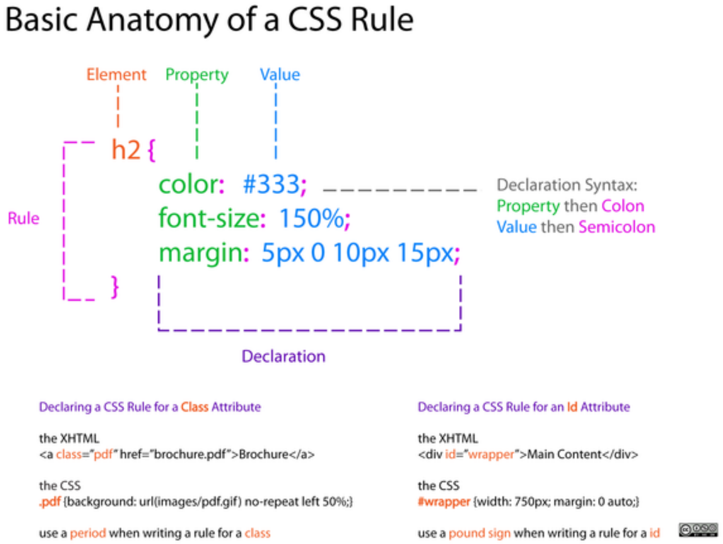
```
<link href="style.css" rel="stylesheet" />
```

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

anatomy of a CSS rule

Each CSS style sheet (all the styles of a page) is made of several rules.

Each **rule** follows the syntax:



Basic Anatomy of a CSS Rule

```
h2 {
    color: #333;
    font-size: 150%;
    margin: 5px 0 10px 15px;
}
```

Declaring a CSS Rule for a Class Attribute

the XHTML
<a class="pdf" href="brochure.pdf">Brochure</a>

the CSS
.pdf {background: url(images/pdf.gif) no-repeat left 50%;}

use a period when writing a rule for a class

Declaring a CSS Rule for an Id Attribute

the XHTML
<div id="wrapper">Main Content</div>

the CSS
#wrapper {width: 750px; margin: 0 auto;}

use a pound sign when writing a rule for a id

Source: http://dabrook.org/resources/posters/

Example of a CSS rule

- **element**: what element(s) is being styled e.g. *div*
- **property**: what property of that element is being styled e.g. *color*
- **value**: how the property is styled e.g. *white*

```
div {
      background: blue;
      color: white;
      width: 500px;
      height: 250px;
      font-size:30pt;
}
```

Here we are styling all the div elements in the html page.

CSS Properties

CSS Property documentation: https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

Some properties:

- color, background-color, width, height
- border, box-shadow, list-style
- margin, padding
- transform, gradient, border-radius

Inspecting a page

The browser offers the possibility of inspecting a page with the option **Inspect Element** or just **Inspect**.

This possibility allows for prototyping (changing and seeing immediately the result) a page's CSS and HTML.

Keep in mind that this changes WILL NOT be saved. To do so you need to copy them to the editor and save them.

CSS selectors

Documentation: https://developer.mozilla.org/en/docs/Web/Guide/CSS/Getting_started/Selectors

CSS selectors allow the selection of html elements to be styled.

Their scope can be very broad, such as all the elements (*), or all the elements that that share a given tag.
To more fine grained selectors, like descendents and id.

To pseudo class selectors, that are triggered by a certain action.

Select a specific element

```
p {
    font-weight: bold;
}
```

Select all (using star)

```
* {
    font-weight: bold;
}
```

Star targets all the elements in a page.

id

#

- id targets the (only 1) element with the given id

- it help *distinguish* elements with the same tag.
- the same id cannot be repeated in the same file. Use only once.

```
h1#title {
    color: pink;
}
```

class

```
.
```

- classes target several elements that share the same class
- classes can be used INFINITE TIMES in a file
- it help *uniforming* different types of elements, or multiple elements that need to be styled in the same way

```
div.article {
    margin-top: 10mm;
}
```

descendents

Descendents are elements that are descendents another element, like the anchors within a list item, and not other anchors

```
li a{
    color: green;
}
```

direct descendents

Direct descendents are elements that **direct children** of another element.

```
li > a{
    color: blue;
}
```

pseudo classes

For example:

- All links that have been visited
- when hovering a link

```
a:visited {
    transform: rotate(0.5turn);
}
a:hover{
    background: red;
}
```

More on pseudo classes: https://developer.mozilla.org/en/docs/Web/CSS/Pseudo-classes

Positioning

Documentation: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Positioning

position:static;

- the default position
- dont offset possibilities
- are positioned according to their default behavior

position: relative;

- very similar to that of the static value
- Main difference: relative value accepts box offset properties top, right, bottom left.
- Box offset properties allow precise positioning

position: absolute;

- elements accept box offset properties (left,right, top, bottom)
- elements are removed from the normal flow of the document
- and positioned in relation to the body element
- off-set property are set in relation to the body and not containing element. E.g. `top: 10px;` will place the element 10px offset from the top of the browser window.

Nice art work using absolute position and Google books image: http://www.julienlevesque.net/books-scapes/

position: fixed;

- similar to absolute: off-set set in relation to the body
- but the **positioning is relative to the browser viewport**
- not scrolling with the page.
- always present, as if fixed to the screen

Display

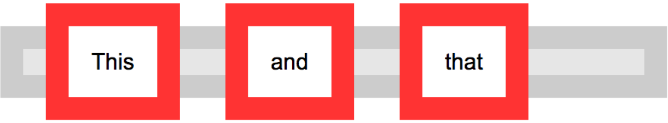
Documentation: https://developer.mozilla.org/en-US/docs/Web/CSS/display

Every element on a web page is a rectangular box.

The CSS display property determines how that rectangular is displayed next to its sibling elements.

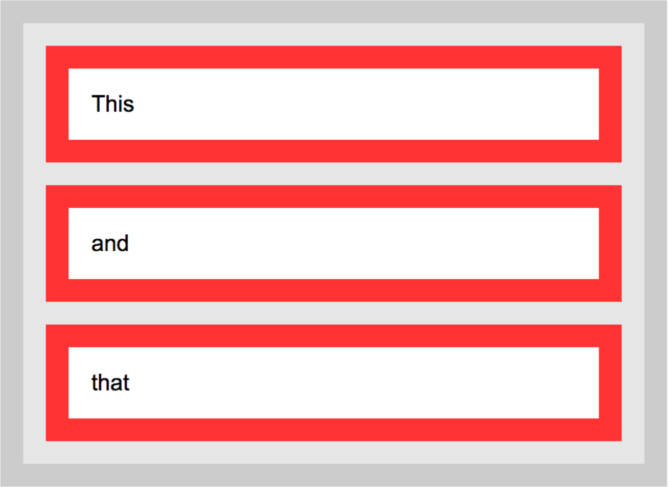
inline

elements are displayed in a line.



block

Each element is standalone, occupying the entire width of its parent box and line breaks before and after it.



inline-block

display the element in a line, like inline, but allows more formatting possibilities: width, height, margin to the right and left of the box.

none

Turns off the display of the element

The default value is inline.

CSS for typography

Common typographic properties

- font-size: *body in pt, rest of elements in em*
- font-height: regular or bold
- font-style
- font-family
- color
- text-align
- line-height
- letter-spacing: increases or decreases the space between characters(negative values are allowed)
- text-shadow

Using fonts

system fonts	**custom fonts**
ready to use	need to load or upload
limited set	broad range
change slightly in each user's computer	remain the same to all users
too familiar	fresh

system fonts

Documentation: https://developer.mozilla.org/en-US/docs/Web/CSS/font-family

System fonts are generic fonts.

- sans-serif
- serif

- monospace

system fonts example

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <style>
body {
   font-size: 10pt;
}

p {font-family: cursive;
   font-size: 3em;
}

p.other {font-family: sansserif}

span.yetanother {font-family: monospace}
    </style>
  </head>
  <body>
    <p>Cursive generict font</p>
    <p class="other">Testig another generic font.
      <span class="yetanother">And yet another one</span>
    </p>
  </body>
</html>
```

custom fonts

The `@font-face` CSS at-rule allows authors to specify online fonts to display text on their web pages.

https://developer.mozilla.org/en/docs/Web/CSS/@font-face

Web fonts formats

Different font formats exist:

- Web Open Font Format (.woff)
- TrueType/OpenType (.ttf/.otf)
- Scalable Vector Graphics Fonts (.svg)

Currently, most browsers support these font formats, with the exception of .svg, that is only supported by Firefox. See Wikipidia article on Web fonts.

use a custom font

To use a custom font, the font file has to be stored somewhere, either **locally** (same folder as your site).

Try changing the following example with other custom font:

```
<html>
<head>
  <style type="text/css">

    @font-face {
      font-family: "Pacifico";
      src: url("fonts/Pacifico.ttf");
    }

    h1 { font-family: "Pacifico", serif;
    font-weight: normal;
    }
  </style>
</head>
<body>
  <h1>This is a custom font.</h1>
</body>
</html>
```

fonts and licenses

We can use this fonts and even make a commercial (for which we receive money) website, using that font, without paying for the font.

It is not because the font is free (of charge), but because they are **open**.

They are released under a **SIL Open Font License** (OPF).

SIL license

http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL_web

License:

- allows fonts to be **used, studied, modified and redistributed freely**.
- fonts, including any derivative works, can be bundled, embedded,

redistributed and/or **sold with any software**.

- fonts and derivatives, however, **cannot be released under any other type of license**.
- requirement for fonts to remain under this license does not apply

to any document created using the fonts or their derivatives.

Apache license

http://www.fontsquirrel.com/fonts/sinkin-sans

open fonts libraries

- https://fontlibrary.org
- http://fontsquirrel.com/
- https://www.design-research.be/by-womxn/
- https://velvetyne.fr/maintenance.html
- https://usemodify.com/

remote custom fonts

It is possible to use custom fonts, that are not stored locally, and instead "live" in a service like Google Fonts or Open Font Library.

This method is:

- convenient
- easier

but, on the down-side:

- it takes more time to load the page
- the font can be removed at any point by the service

remote custom font example

Using the font https://fontlibrary.org/en/font/barrio and following the "Use this font" instructions:

```
<html>
<head>
<link rel="stylesheet" media="screen" href="https://fontlibrary.org/face/barrio" type="text/css"/>

  <style type="text/css">

    h1 {   font-family: 'BarrioRegular';
           font-weight: normal;
           font-style: normal;
           font-size: 3em;
    }

  </style>

</head>
<body>
  <h1>This is a remote custom font.</h1>
```

```
</body>
</html>
```