Print Stylesheets

Although it is possible to print web pages directly there are a few things we can do to make the experience better. We'll explore changes we can make to a regular stylesheet to make it work better in print.

When it comes to print there are a few things I like to do to make the reading experience better.

- **Remove Background Images and Colors** to make printing easier and not run through ink or bleed text and background colors together
- **Make text size slightly bigger** so it's easier to read in paper where we don't have an option to zoom inn
- **Make margins bigger** to avoid having text flush against the page border
- **Adding refereces to external links** to provide context for the links on the page
- **Remove unnecessary content from the page**
- **Remove images that are not essential to the content** to conserve ink and paper and not distract the reader
- **Remove videos or make sure that they display a poster image** to avoid printing things that are not needed

The first three items on the list can be done on the print style sheet; the other two need more manual touch, maybe adding a class to images that are not essential and using that to style the elements accordingly.

Linking to the style sheet

When working with print-specifc styel sheets we use the link element's `media` attribute to specify what medium we want the style sheet to work with.

```
<link rel="stylesheet" media="print" href="print.css">
```

Most of the time I see

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

But we forget that printing a page and viewing it on a screen will have very different results and not all of them are predictable for us. Instead use two declarations like this:

```
<link rel="stylesheet" media="screen,projection,tv" href="styles.css">
<link rel="stylesheet" media="print" href="print.css">
```

The decalrations will pass the screen style sheet to devices that are screen-based and print style sheets that support the printed rules.

Remove Background Images and Colors

The first I do is to remove background image and background color from the page. I also set the page color to black.

```
body {
    background-color: #fff;
    background-image: none;
    color: #000;
}
```

We do this to save ink and prevent the colors from bleeding into each other.

Make text size slightly bigger

When working with print I like to use points instead of pixels and make sure that I explicitly stick to the 16px/16pt default from the browser.

```
body {
    font-size: 16pt;
}
```

Make margins bigger

To make sure sure that content doesn't stick too close to the borders of the page, I like adding extra margins to the body element. You can tweak the margin value to give more or less space to the content.

```css
body {
  margin: 2em;
}
```

Adding references to external links

Printed style sheets will not tell you what a link represents and since we've removed the underline from links there is no easy way to distinguish it from regular text.

We're doing three things:

- We remove links underline and make them black
- We create a text attribute with the value of the link's `href` attribute
- We set links so that they will break at the word boundary rather than let the link go beyond the width of the page

```css
a, a:visited {
color: #000;
text-decoration: none;
}

a[href^="http"]:after {
content: " (" attr(href) ")";
word-wrap: break-word;
}
```

Using these rules the link:

```html
<a href="https://twitter.com/">Twitter</a>
```

will appear on the page as:

Twitter (https://twitter.com)

Cleanup video contents

Video content in printed media can be problematic and it'll depend on how you add the video to your page. The easiest way (in terms of storage and use) is to link directly from Youtube or Vimeo.

The Vimeo links look like this:

```html
<div class="video">
  <iframe width="560" height="315" src="https://www.youtube.com/embed/MY3
</div>
```

Youtube looks like this:

```html
<div class="video">
  <iframe width="560" height="315" src="https://www.youtube.com/embed/MY3
</div>
```

One way to work with this is to hide the iframe and put a link to the video inside in a similar to what we did with links. We can then style them as needed.

```css
iframe {
  display: none
}

iframe[src*="youtube"]:after,
iframe[src*="vimeo"][src*="vimeo"]:after {
content: " (" attr(src) ")";
word-wrap: break-word;
}
```

This will not work when hosting video locally and referencing it with a video tag, either containing regular video or a DASH manifest.

I'm making a few assumptions. When not linking from Youtube or Vimeo, I use the following structure for the video:

```
<div class="video">
  <video
    width="560"
    controls
    poster="path/to/video1-poster.jpg" >
      <source src="https://example.org/video/video-high.mp4" type="video/
      <source src="https://example.org/video/video-medium.webm" type="vide
  </video>
</div>
```

In this case we would remove the video element and add the link from the first source attribute denoted with the `:first-child` pseudo attribute (we do this in case therre is more than one source child for the video. The CSS now looks like this:

```
video {
  display: none
}

source:first-child:after {
content: " (" attr(src) ")";
word-wrap: break-word;
}
" ("
```

Remove unnecessary content from the page

The final tweak to a page is to remove stuff we don't need to have to read the page; for example social links, ads and other stuff that will not affect the reading experience.

For example, let's assume that we have a div with class social and a div with

class ads that we want to get rid of in a print stylesheet. Removing the element is as simple as setting `display: none` like so:

```css
.social,
.ads {
  display: none;
}
```

The complete style sheet (so far)

This is the style sheet that contains all the rules we've e disc ussed so far. It's not a finished product and you will have to decide what else to take out and how best to do it.

```css
body {
  background-color: #fff;
  background-image: none;
  color: #000;
  font-size: 16pt;
  margin: 2em;
}

a, a:visited {
color: #000;
text-decoration: none;
}

a[href^="http"]:after {
content: " (" attr(href) ")";
word-wrap: break-word;
}

" ("iframe[src*="youtube"]:after,
iframe[src*="vimeo"][src*="vimeo"]:afterontent: " (" attr(src) ")";
word-wrap: break-word;
}
```

```
vide" ("video {
    display: none
}

source:first-child:afternt: " (" attr(src) ")";
word-wrap: break-word;
}

.social," (".social,
.ads {
    display: none;
}
```

You may also have to tweak some of the rules and selectors below. For example, I've only cleared the background image on the body and not considered that other elements inside the body of the page may also have their own backround color and/or images.

Still this is a good beginning :)