# A sample page

## Relevant Links

- Some layout techniques[1]
- Flexbox Layout[2]
- Color picker[3]
- Color pallette generator[4]
- CSS units[5]
- W3C's section on CSS units[6]
- Safe web fonts[7]
- Google fonts[8]
- Font scaling[9]

## Notes

In this section we will provide an example of a web page and some styling of it. We start with the HTML for the page.

```
<!doctype html>
<html>
  <head>
    <title>A sample page</title>
    <link rel="stylesheet" type="text/css" href="reset.css">
    <link rel="stylesheet" type="text/css" href="basics.css">
  </head>
  <body>
    <header>
      <img id="logo" src="https://www.hanover.edu/images/site/logos/logo.png" alt="Our␣colleg
      <h1>An awesome sample page!</h1>
    </header>
    <main>
      <section id="sidebar">
        <nav>
          <h2>Navigation</h2>
          <ul>
            <li><a href="page1.html">A link!</a></li>
            <li><a href="page2.pdf">Another link!</a></li>
            <li><a href="page2.pdf">A third link!</a></li>
          </ul>
        </nav>
      </section>
```

[1] http://web.simmons.edu/~grabiner/comm244/weeknine/css-layouts.html
[2] https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox
[3] http://htmlcolorcodes.com/color-picker/
[4] http://paletton.com/
[5] https://www.tutorialspoint.com/css/css_measurement_units.htm
[6] https://www.w3.org/Style/Examples/007/units.en.html
[7] http://web.mit.edu/jmorzins/www/fonts.html
[8] https://fonts.google.com/
[9] http://type-scale.com/

```
      <section id="content">
        <h2>Our awesome content!</h2>
        <article>
          <h3>First item</h3>
          <p>Cool content goes in this paragraph. A lot of stuff we can write here.</p>
          <p>Another paragraph within the same item.</p>
        </article>
        <article>
          <h3>Second item</h3>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur pharetra scel
          <p>Donec auctor tincidunt felis nec sollicitudin. Mauris scelerisque in mauris nec
          <p>Morbi ac convallis dui. Cras sed leo ut dui tempus eleifend. Vivamus ac ipsum at
        </article>
      </section>
    </main>
    <footer>Thank you for visiting!</footer>
  </body>
</html>
```

We start by saving this into a file, and open it up on a browser. It probably doesn't show at all like we want it to, even though we have semantically described the content well. Let's look at some of the problems:

- The header section is not visibly separated from the rest.
- The logo, which presumably was supposed to be much smaller and to a side, takes up a lot of space.
- The top heading was probably meant to be to the right of the logo, not below it.
- The navigation bar was meant to be to the left, with all other content to its right.
- The footer isn't visibly distinct from the rest.
- The two "articles" kind of run into each other without much visible separation.
- The navigation links are somewhat boring normal links. We'll change them into clickable blocks.

### Cleanup / CSS reset

A common first step in any page design is eliminating all browser settings. This is often called a CSS-reset. This often ends up being somewhat lengthy. We put it into its own file (reset.css) that loads first.

```
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
```

```
    margin: 0;
    padding: 0;
    border: 0;
    font-size: 100%;
    font: inherit;
    vertical-align: baseline;
}
/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
    display: block;
}
body {
    line-height: 1;
}
ol, ul {
    list-style: none;
}
blockquote, q {
    quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
    content: '';
    content: none;
}
table {
    border-collapse: collapse;
    border-spacing: 0;
}
```

Look at the page after this change. You will see it now looks a mess. But we'll clean it up.

The rest of the CSS changes will all go into our basics. css file. Though we could in theory have separated some of the changes into different files (for instance of some changes are to be shared with other websites or pages).

### Logo adjusting

We start by taming in the logo a bit.

```
#logo {
    height: 2em;
    width: auto;
    padding: 1px;
    background-color: black;
    border: 1px solid gray;
}
```

So:

- We give it a fixed height, set to 2 "ems". An em is a measurement unit representing the font size. Values specified in ems will scale if you zoom in/out of the page.

- We give it a width of auto, meaning that it will maintain the image's width/height ratio. Otherwise the image would appear distorted.
- We add a 1px padding around the image.
- We set the background color of the image to be black (we will use a different background color for the rest of the header in a moment).
- We put a slight border to it. This is a "compound rule", where we set multiple border properties at once (border-size, border-style, border-color).
- We have specified the color here by name. A number of colors can be specified that way. We can also specify colors via RGB values, and we will do so in the rest.

**Basic layout**

The first big item is specifying the basic layout that will create the different page sections.

```
header {
  background-color: #D9ECC3;
  padding: 10px;
}

main {
   display: flex;
   flex-direction: row;
}

#sidebar {
   flex: 1;
   padding: 1em;
   background-color: #799657;
}

#content {
   flex: 3;
   padding: 1em;
   background-color: #EFF4E9;
}

footer {
   border-top: 1px solid #4A523F;
   padding: 10px;
}
```

So here is what is going on:

- The main regions are all given background colors.
- The sidebar, content, header and footer receive some padding, so that text does not hit the boundaries.
- The "main" section is set to "display" as "flex" and with a "flex-direction" set to row. This means that its children should arrange themselves horizontally next to each other, forming columns.

- The sidebar and content, which are the children of "main", are given "flex" values that indicate their relative size. In this case we want the content to be three times as large as the sidebar.

At this point, let's go ahead and add: `display: flex;` to the header rule. This will allow the image and header text to stand next to each other.

### Typography

We will now do some styling of the typography of the various sections.

```
/* Typography */
body {
    font-family: Georgia, Palatino, "Times New Roman", serif;
    font-size: 16px;
    line-height: 1.2;
}

h1, h2, h3 {
    font-family: Verdana, Tahoma, Helvetica, sans-serif;
    font-weight: bold;
}
h1, h2 { text-align: center; }
h1 { font-size: 2em; }
h2 { font-size: 1.414em; }
h3 { font-size: 1em; }
```

So here is what these lines do:

- We set a basic font family in the body, that will be inherited by other tags contained in the body. The font family value is a list of different choices. The browser will try to apply the left-most choice if it is present in the system. The last choice, `serif`, tells the browser to use its default serif font.
- You could also link to many other kinds of fonts, like the Google fonts linked to at the top of the page. You can then use those instead of the standard ones.
- We then set the base font size to be 16 pixels. Users can adjust that when they increase the font on a page (most browsers let you do that). This setting specifies the meaning of an "em" for the rest of the document.
- We overwrite the font family for header items, to be one of the `sans-serif` fonts.
- We set the text aligning for the top two headings to be centered.
- We specify the size of the heading fonts to be relative to the base setting.
- The `line-height` setting specifies that the distance between lines should be about 20% larger than the standard character height. This spaces the lines out a bit and makes them more readable.

### Spacing

Let us work on spacing the items on the content section a bit, and learn some more tricks along the way:

```
/* Content area */
main section {
   padding-top: 0.5em;
}

article {
   margin: 1em;
   padding: 0.2em;
   border: 1px solid #EFF4E9;
   border-radius: 7px;
}

article p {
   margin-bottom: 0.3em;
   margin-top: 0.3em;
}

article p:first-of-type::first-letter {
   font-size: 1.414em;
}

article p:last-of-type {
   border-bottom: 1px dashed #999998;
}

article:hover {
   background-color: #E1E6DC;
   border-color: gray;
}

article:hover h3:after {
   content: '\25ca';
   padding-left: 0.2em;
   color: #A2EDF3;
}
```

Let's walk through what each of these settings does:

- We start by applying a top margin to all section elements that are within the main element. This applies to the sidebar and the content. We could also have targeted those two as #sidebar, #content. This moves the h2 headings a bit down so they don't brush up against the header.
- We then style each article element. We add some margin around them so they can be clearly separated from each other, and some padding that together with a border will make a nice hover effect. We then do a bit of magic, setting a border with the same color as the background. We will change the color later, but this allocates space for the border and prevents weird movement later on. Lastly, we set the "border-radius" property, which controls how rounded the corners of the boundary will be.
- Next we add a bit of margin for the paragraphs within an article. Note that each element has its top margin as well as the bottom margin of its previous sibling, but only one of the margins applies. Padding would not have worked as well,

leaving either too much gap inbetween paragraphs or too little spacing at the beginning.

- On the next item we apply some style to the "first letter of the first paragraph in any article". That style makes that letter of bigger font size.
- The next item shows how we can add something to the last paragraph of each article.
- Next we add an effect when we hover over an article's area. We change the background and add a border color (remember we already allocated space for the border earlier; if we hadn't, then there would be more movement of elements when we hover over something).
- Finally, We set up an effect to happen to any h2 element within an article element that is being hovered. More specifically, we are defining something to go right after the h2 element, and the content to be added there is the string with a single character with unicode number 25ca. We also specify a color for this added content, and some spacing.

### The sidebar

The time has come for us to work on the sidebar. The sidebar contains an unnumbered list of list items, each of which is a link. We'll create rectangular clickable areas for each of them.

```
/* sidebar */
#sidebar nav {
   margin: auto;
   width: 80%;
}

#sidebar nav li {
   margin-top: 0.2em;
   margin-bottom: 0.2em;
}

#sidebar nav li a {
   display: block;
   text-align: center;
   text-decoration: none;
   color: black;
   border: 1px solid #5E8251;
   border-radius: 3px;
   background-color: #6DA52A;
}

#sidebar nav li a:hover {
   border-color: #4D6A42;
   background-color: #578322;
}

#sidebar nav li a[href$=".pdf"]:after {
   background-image: url("http://iconbug.com/data/5b/507/52ff0e80b07d28b590bbc4b30befde52.png
   background-size: 1em 1em;
   display: inline-block;
```

```
    width: 1em;
    height: 1em;
    content: "";
}
```

Let's see what goes on here:

- We vertically align the nav element by giving it a fixed (computable) width and having it automatically arrange the margin.
- We add margins to the list items to space them out.
- The links within the list items turn into block elements, and we remove the normal link decoration (underline).
- On a hover over a link, the border and background colors change.
- The last bit of crazyness adds an image to the right of links that end in a ".pdf". It does so in a weird way, by specifying a content of an empty string, but with given width and height and a background image that is a resized icon.