

Page Layout

After completing this section, you will:

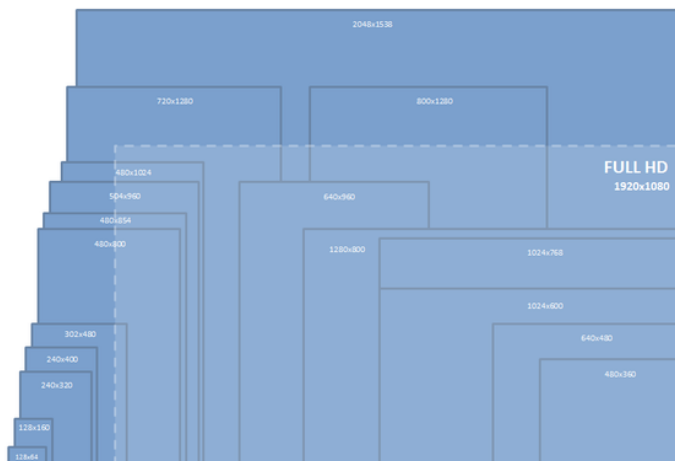
- ☐ Identify the purpose of normalize.css documents
- ☐ Be able to use a container and center your page
- ☐ Create Headers and Footers
- ☐ Use Float to Create Columns
- ☐ Use Media Queries to create a more responsive design

Introduction

Page layout with CSS is complicated by the fact that the destination screen size and resolution are most often unknown. One viewer could view your content on a large screen monitor, while others might be using a mobile device screen.

Despite the diversity of screen sizes, it's essential that digital content remains both readable and attractive regardless of the environment in which it's viewed. The image below illustrates the diversity of screen sizes and the challenge that digital designers often face.

In this section, we're going to look more closely at a couple of common page layout techniques.



Normalize.css

We'll begin with the HTML5 basic document structure with one addition. Key in the following code while making special note of the URL:

```
<!DOCTYPE html>
<html>
<head>
<link href="https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/
normalize.min.css" rel="stylesheet" />
<title>Page Layout</title>
</head>
<body>
</body>
</html>
```

You, of course, noticed the `link` tag added to the `head` section of the document. The CSS style sheet we're linking to is known as a "reset" or "normalize" style sheet. These style sheets are designed to ameliorate the small difference in the default style sheet in various browser. Reset style sheets are also designed to remove some of the unpredictability from design as most unpredictable margin and padding values are set to zero.

The Page Container

When doing more complex page layout, it's a good idea to use a master container `div` to layout your page. This is a box that surrounds all of your other content. The page container `div` allows you to attach styles to all content on the page without using the `body` tag as a CSS selector.

The page container makes a better parent for elements on the page than the `body` tag. We'll implement the page container inside the `body` tag.

```
<!DOCTYPE html>
<html>
<head>
<link href="https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/
normalize.min.css" rel="stylesheet" />
<title>Page Layout</title>
</head>
<body>
  <div id="container">
    </div> <!-- container -->
</body>
```

Now that we have our container `div` we'll build everything inside this element.

Centering the Container

I want to set up this layout so that the entire design is centered in the browser window. We'll have the layout occupy 90% of the width of the window. We'll also color the background a very light gray, add some padding and set the default typography.

Since the container is our root element, it makes sense to do this here. Centering the content is accomplished by setting the left and right margin to `auto`. (We set the top and bottom margins to 0 in this document.)

Add the following code to the `head` section of your document.

```
<style>
  #container
  {
    width: 90%;
    background-color: #eee;
    margin: 0 auto;
    padding-left: 3px;
    padding-right: 3px;
    font-family: Verdana;
    font-size: 1.1em;
  }
</style>
```

CSS Set Up

We're going to divide our document into several divisions. The body of the document contains the sections and dummy text.

```

<body>
  <div id="container">
    <header><h1>Connecticut News</h1></header>
    <div id="colA">
      <h2>Connecticut Cold in the Winter</h2>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec auctor risus lorem. Nulla consequat ultrices
        <br>
        <p>Nam lacinia feugiat pulvinar. Aliquam erat volutpat. Sed cursus volutpat justo, ac convallis nunc luctus veh

    </div>
    <div id="colB">
      <h2>Connecticut Sun: Pro Basketball at Mohegan Sun</h2>
      <p>In tincidunt tellus metus, vel condimentum diam rhoncus ut. Nullam eu turpis ligula. Nulla vehicula nisl aug

      <p>Nulla sed mauris auctor, sagittis ipsum quis, aliquam mauris. Pellentesque lobortis faucibus hendrerit. Sed
    </div>
    <div id="colC">
      <h2>Mysteries of Connecticut</h2>
      <p>Donec eget euismod orci, a ultrices tortor. Ut et pretium arcu. Ut ut est id elit dapibus aliquam ut eu mass
    </div>
    <footer>Copyright &copy;2018 Framework Television</footer>
  </div> <!-- container -->
</body>

```

If you want to use dummy text for this exercise (or for anything else) you can copy it from www.lipsum.com.

You'll notice that we use the id's `colA`, `colB`, and `colC` to label our individual columns. We also have header and footer tags that contain (obviously) the page header and footer content.

We'll create entries in our `<style>` element to hold the CSS we'll need to create for each of the content areas.

```
<style>

    #container
    {
        width: 90%;
        background-color: #eee;
        margin: 0 auto;
        padding-left: 3px;
        padding-right: 3px;
        font-family: Verdana;
        font-size: 1.1em;
    }

    header
    {

    }

    footer
    {

    }

    #colA
    {

    }

    #colB
    {

    }

    #colC
    {

    }

</style>
```

Even though we haven't placed the style rules in the document yet, we can see the structure starting to take shape. Creating placeholders for style rules is especially helpful when you have complex documents with many divisions and content areas.

Note that the `#container` rules come first in the style sheet. This is by design. Any individual rules in the content areas that differ from the container override the original container rules.

Header and Footer

Most websites have a header and footer content. (Sometimes header and footer content can be extensive)

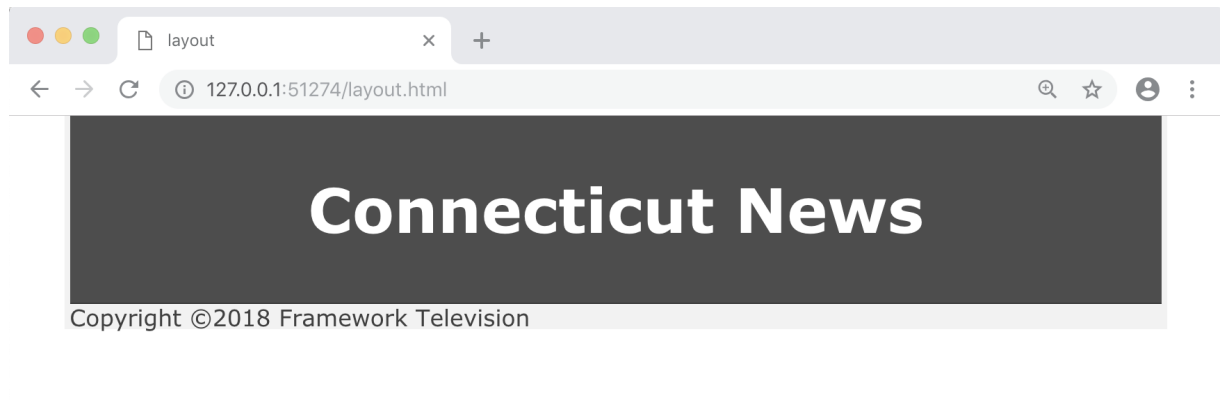
For the `header` element, let's center the content, reverse the background and text colors and add some padding.

```
header
{
    text-align: center;
    background-color: #121212;
    color: white;
    padding-top: 10px;
    padding-bottom: 10px;
}
```

For the `footer` element, let's add a line above it and reduce the size of the type.

```
footer
{
    border-top: 1px solid black;
    font-size: .75em;
}
```

With the few styles listed above, we can already get a sense of what our header and footer will look like in the browser. (The line above the footer isn't visible because it's right up against the header section.)



Column Floats

The design we're working on is a three-column layout. We'll accomplish by setting appropriate column widths, float rules, and borders. We'll also add some padding so that the text content from different columns don't crowd and become hard to read.

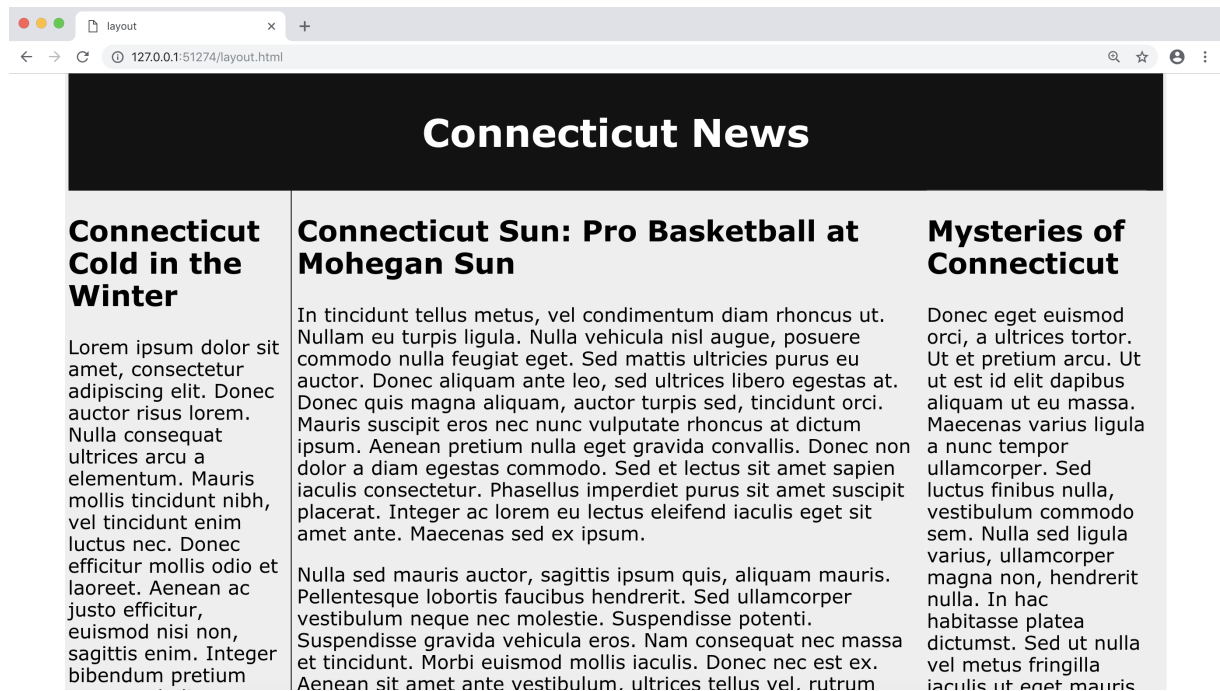
Let's add the following rules to our `style` element:

```
#colA
{
    width: 20%;
    float: left;
    border-right: 1px solid black;
    padding-right: 3px;
}

#colB
{
    width: 57%;
    float: left;
    padding-left: 5px;
    padding-right: 5px;
}

#colC
{
    width: 20%;
    float: left;
}
```

If you test in the browser, you should see your content laid out in three columns.



However, if you scroll down, you may notice a problem with the footer.



When you float an element in the browser you “break” the natural stacking of elements. The result is that any elements below the floated elements are laid out somewhat unpredictably. We can fix this by clearing the floats in the footer element:

```
footer
{
  border-top: 1px solid black;
  font-size: .75em;
  clear: both;
}
```

Once you add the `clear` rule to the footer, the problem will rectify itself. Now, our layout looks pretty good on a laptop screen. However, we have some problems on smaller screens. If the screen isn't wide enough to accommodate the three columns, and content, things get crowded quickly.



When the screen isn't wide enough to accommodate the content, the result is unacceptable. Generally, we'll undo the floats and stack our layers on smaller screens to make our content readable.

We accomplish this with media queries.

Media Queries

Media queries allow us to create breakpoints at which the design changes based on specific rules, usually on document width. Let's add the following media query to the bottom of our `style` element.

```
@media only screen and (max-width: 600px)
{
    #colA, #colB, #colC
    {
        float: none;
        width: 100%;
    }

    #colA
    {
        border-right: none;
    }
}
```

This media query states that the included rules will apply up to a maximum width of 600px. The float is removed from the three columns, and the border on the right side of `#colA` is removed as well. Any browser that is wider than 600px will not apply these rules.

Many developers create different media queries for phones, tablets, and large screens. Others even target specific models of phones and tablets with their media queries. Going overboard is easy! In general, you want your media queries to ensure your content is accessible across a wide array of screen sizes.

Do This

Using document we've created redo the media Queries based on typical iPhone and iPad width. You can find information about Apple display sizes at

<https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/Displays/Displays.html>.

Please note that it's easy to overdo it with media queries. Just add one or two queries to address a few Apple screen sizes.

Note: You can download the correct version of the document at

<https://s3.amazonaws.com/coursewareframework/html5authoring/layout.html.zip>

Submit This

Download the lab starter at <https://s3.amazonaws.com/coursewareframework/html5authoring/LayoutLab.zip>. View the content in your browser. It should appear similar to the image below.



News About Nothing

All the Nothing News That's Fit to Post

Airliner Takes Off, Lands Normally

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque elit metus, euismod ac lorem sit amet, accumsan faucibus odio. Curabitur interdum nec dolor nec ultricies. Sed eget erat et est tincidunt mattis id ut nisl. Nullam nec efficitur enim. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Quisque finibus tincidunt elementum. Proin dolor eros, molestie sit amet aliquam in, feugiat et nulla. In vitae posuere purus. Sed malesuada quis leo eget mollis. Aliquam blandit massa vitae molestie efficitur.

Fusce convallis sollicitudin finibus. Integer ac ipsum tincidunt, malesuada urna consequat, scelerisque augue. Etiam et eleifend metus. Vestibulum quam elit, ullamcorper ut mi imperdiet, vehicula euismod sem. Quisque consequat pellentesque lorem eu elementum. Sed molestie risus ut turpis porttitor, ut faucibus lacus ornare. Etiam quis dolor volutpat, porta libero sed, dignissim felis. Mauris blandit eu ex ac posuere.

Sed vel turpis efficitur, vulputate dui ac, consectetur sem. Proin ornare ullamcorper ornare. Aliquam a turpis tortor. Aenean nisl nibh, semper ut consectetur ac, feugiat ut quam. Sed eu dolor felis. Cras et ligula sodales, vehicula mauris imperdiet, tempus elit. Vivamus id neque consequat, lacinia tortor in, dapibus magna. Sed commodo orci id nisl scelerisque, quis varius nisl efficitur. Maecenas pretium lobortis dictum. Aliquam nunc ante, gravida tempus risus venenatis, imperdiet pellentesque ex.

Worker Goes to Work and Comes Home

Brave Worker Calls it "Just Another Day"



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque elit metus, euismod ac lorem sit amet, accumsan faucibus odio. Curabitur interdum nec dolor nec ultricies. Sed eget erat et est tincidunt mattis id ut nisl. Nullam nec efficitur enim. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Quisque finibus tincidunt elementum. Proin dolor eros, molestie sit amet aliquam in, feugiat et nulla. In vitae posuere purus. Sed malesuada quis leo eget mollis. Aliquam blandit massa vitae molestie efficitur.

Fusce convallis sollicitudin finibus. Integer ac ipsum tincidunt, malesuada urna consequat, scelerisque augue. Etiam et eleifend metus. Vestibulum quam elit, ullamcorper ut mi imperdiet, vehicula euismod sem. Quisque consequat pellentesque lorem eu elementum. Sed molestie risus ut turpis porttitor, ut faucibus lacus ornare. Etiam quis dolor volutpat, porta libero sed, dignissim felis. Mauris blandit eu ex ac posuere.

Sed vel turpis efficitur, vulputate dui ac, consectetur sem. Proin ornare ullamcorper ornare. Aliquam a turpis tortor. Aenean nisl nibh, semper ut consectetur ac, feugiat ut quam. Sed eu dolor felis. Cras et ligula sodales, vehicula mauris imperdiet, tempus elit. Vivamus id neque consequat, lacinia tortor in, dapibus magna. Sed commodo orci id nisl scelerisque, quis varius nisl efficitur. Maecenas pretium lobortis dictum. Aliquam nunc ante, gravida tempus risus venenatis, imperdiet pellentesque ex.

Phasellus enim turpis, vehicula et lorem quis, auctor accumsan augue. Donec ultricies tortor vitae erat vestibulum maximus. Aliquam et pellentesque ligula. Nullam ligula erat, luctus a lacus eget, ultrices venenatis purus. Nulla non sapien vel lectus venenatis vehicula. Morbi at congue est, eget congue dolor. Cras tincidunt lorem a auctor lacinia. Phasellus euismod velit in tellus tincidunt, sed luctus ex egestas. Donec iaculis faucibus mauris sit amet pretium. Nunc eget velit efficitur, efficitur neque tempus, suscipit magna. Aliquam consequat dolor et lacus rutrum vehicula. Quisque tristique viverra orci vitae mattis. Sed luctus est odio, ut ultrices velit elementum nec. Donec ultricies tempus felis, sit amet bibendum dui tempor id. Pellentesque in varius nunc.

Nulla ultrices, diam ut rhoncus luctus, libero lorem ultrices quam, non suscipit augue neque cursus ex. Nullam cursus volutpat pretium. Etiam eu enim commodo, cursus ante ac, venenatis lacus. Etiam non accumsan quam. Integer sit amet enim sit amet nibh commodo posuere. In semper malesuada ex nec pulvinar. Morbi vel mauris porttitor, dictum nisl quis, efficitur metus. Donec auctor nisi mauris. Vestibulum ut lorem nulla. Vivamus porta aliquet hendrerit. Phasellus ultrices, odio condimentum sagittis tempor, leo enim lobortis leo, eget tempor lacus erat non purus. Suspendisse id nisi in dui euismod auctor mattis ut neque. Suspendisse potenti.

Hairbands: Out in the 80's, Back in the 2010's

Phasellus enim turpis, vehicula et lorem quis, auctor accumsan augue. Donec ultricies tortor vitae erat vestibulum maximus. Aliquam et pellentesque ligula. Nullam ligula erat, luctus a lacus eget, ultrices venenatis purus. Nulla non sapien vel lectus venenatis vehicula. Morbi at congue est, eget congue dolor. Cras tincidunt lorem a auctor lacinia. Phasellus euismod velit in tellus tincidunt, sed luctus ex egestas. Donec iaculis faucibus mauris sit amet pretium. Nunc eget velit efficitur, efficitur neque tempus, suscipit magna. Aliquam consequat dolor et lacus rutrum vehicula. Quisque tristique viverra orci vitae mattis. Sed luctus est odio, ut ultrices velit elementum nec. Donec ultricies tempus felis, sit amet bibendum dui tempor id. Pellentesque in varius nunc.

Create a traditional three column layout with header and footer using the HTML document provided. Using media queries, ensure that your document works across a variety of browser widths.

Remember, when submitting the work please use the following naming convention for your file:

HTMLAUTHORING_LastName_SectionName.html . So if your last name is Smith and you are submitting this section. Your file name should be HTMLAUTHORING_Smith_Layout.html . Since you have two files for this exercise, please zip them together before uploading.

For this course go to <https://www.dropbox.com/request/RhW9kBDXtisq2Fsvg3hY> to submit your assignments.