

Introduction to CSS

CSS Files

What is CSS

- **Cascading Stylesheets**
- NOT a programming language
- Stylesheet/Styling language
- Used for website layout and design
- Can be extended with Sass/Less

Three main ways to add CSS to html file

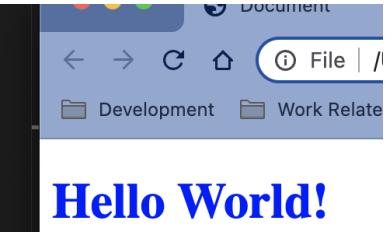
- **Inline CSS:** Directly in the html element (NO!)
- **Internal CSS:** Using <style> tags within a single document
- **External CSS:** Linking an external .css file

Inline (Do NOT do this)



```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Document</title>
</head>
<body>
  <h1 style="color: red">Hello World!</h1>
</body>
</html>
```

Internal (Avoid)



```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    h1 {
      color: blue;
    }
  </style>
</head>
<body>
  <h1>Hello World!</h1>
</body>
```

External (preferred)

First, create a folder called styles and create a new css file called styles.css

```
✓ styles
```

```
# styles.css
```

Inside this file, add a style

```
styles > # styles.css > ...
```

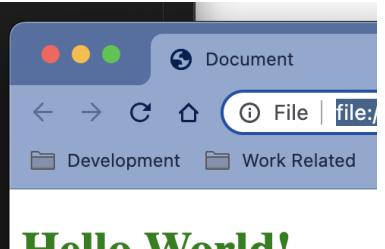
```
1
2
3   h1 {
4     color: green;
5   }
6
```

Then, in the index.html file - in the header section we want to include our stylesheet 'styles.css'

```
<head>
  <title>Document</title>
  <link rel="stylesheet" href="styles/styles.css">
</head>
```

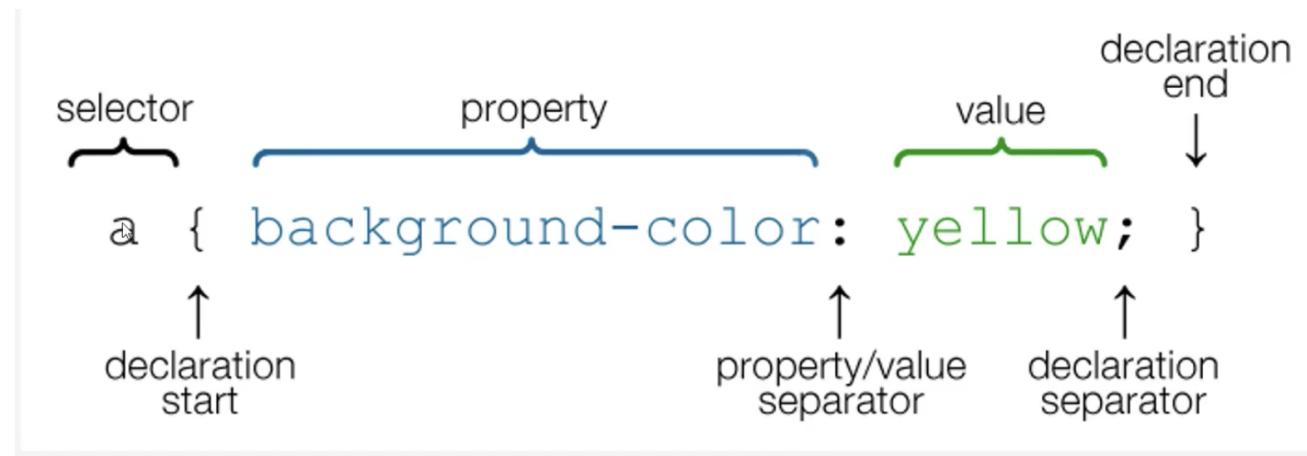
Now, our html file will load the styles and apply them to our html elements we define

```
<html lang="en">
<head>
|   <title>Document</title>
|   <link rel="stylesheet" href="styles/styles.css">
</head>
<body>
|   <h1>Hello World!</h1>
</body>
</html>
```



Hello World!

CSS Selectors



Changing everything in the body of our html

- change the back ground of the page to a soft white with dark grey text

A screenshot of a code editor showing a CSS rule for the `body` element. The rule sets the color to `#555` and the background color to `#f4f4f4`. To the right of the editor, there is a preview area showing the text "Hello World!" in a dark grey font on a light grey background.

```
body {  
    color: #555;  
    background-color: #f4f4f4;  
}
```

Development Work Related

Hello World!

Quick work about fonts: Web safe fonts and external fonts. External fonts need to be included in the header like style sheets

Web Safe Fonts



Examples	Font Families
font-family	Arial, Helvetica, sans-serif
font-family	"Arial Black", "Gadget", sans-serif
font-family	"Bookman Old Style", serif
font-family	"Comic Sans MS", cursive, sans-serif
font-family	Courier, monospace
font-family	"Courier New", Courier, monospace
font-family	Garamond, serif
font-family	Georgia, serif
font-family	Impact, Charcoal, sans-serif
font-family	"Lucida Console", Monaco, monospace
font-family	"Lucida Sans Unicode", "Lucida Grande", sans-serif
font-family	"MS Sans Serif", Geneva, sans-serif
font-family	"MS Serif", "New York", sans-serif
font-family	"Palatino Linotype", "Book Antiqua", Palatino, serif
φωντ-φαμιλίψ	Συμβολή, σανσ-σεριφέ
font-family	Tahoma, Geneva, sans-serif
font-family	"Times New Roman", Times, serif
font-family	"Trebuchet MS", Helvetica, sans-serif
font-family	Verdana, Geneva, sans-serif

Changing the font of the entire page (body)

```
body {
    color: #555;
    background-color: #f4f4f4;

    font-family: Arial, Helvetica, sans-serif;
    font-weight: normal;
    font-size: 16px;
}
```

Hello World!

Ipsum dolor sit amet, consectetur adipisicing elit. Molestiae porro praesentium nostrum consequuntur sunt illo aliquam, hic claudantium enim nam quod labore id culpa pariatur. Cupiditate, repudiandae.

We can also use a shot cut instead of the three lines

```
font: normal 16px Arial, Helvetica, sans-serif;
```

Classes and Id's

CSS styles can use either a class or an ID as the 'Selector'

The selector for a class is the .className

The selector for an ID is the #id

Element ID's should be unique on a page whereas classes can be duplicated multiple times on a page.

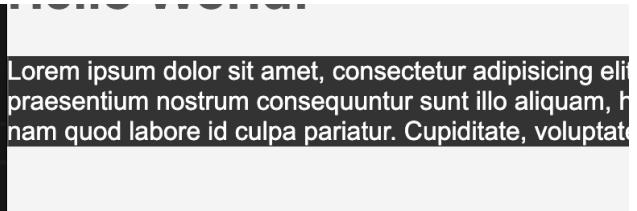
Working with selectors

First, lets wrap some html with a DIV and add a class attribute to it

```
<h1>Hello World!</h1>
<div class="header-card">
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
  | Molestiae iusto preferendis cum excepturi velit porro praes
  </p>
</div>
```

In the css...

```
.header-card {
  color: ■#fff;
  background-color: □#333;
}
```



You can see that the page is going edge to edge... typically, we place everything into a container to give the site a buffer on the left and right. As follows:

In the html, we can wrap everything within the body into a DIV and give it a class of container
*** shorthand emmet: type **div.container** and then press tab

```
<div class="container"></div>
```

Format it as follows

```
<body>
  <div class="container">
    <h1>Hello World!</h1>
    <div class="header-card">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
      | Molestiae iusto preferendis cum excepturi velit porro praes
      </p>
    </div>
  </div>
</body>
```

Now in our css, lets add the container class and view what this looks like

```
.container {  
    width: 90%;  
    margin: auto;  
}  
  
.header-card {  
    color: ■#fff;  
    background-color: □#333;  
}
```

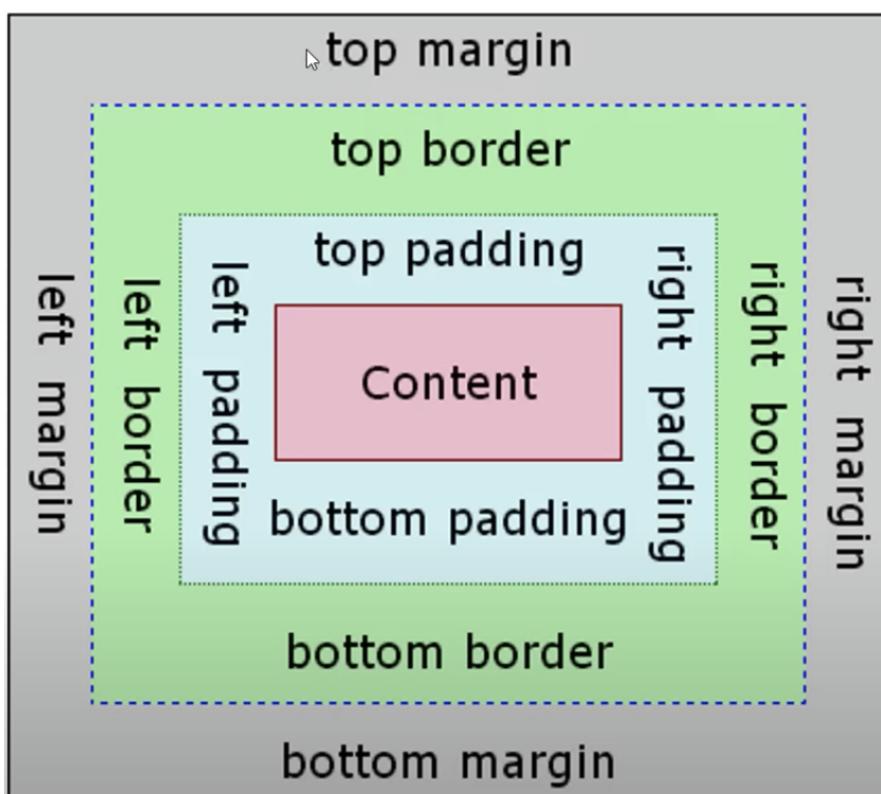


Consequatur, repellendus repudiandae.

Margin Auto will add the **margin:auto** this will make the spacing (created by the width: 90%) equal on either side. Using percentages is better than hard numbers and will allow the page to resize properly.

Element Border, margins, and Padding

Box model



Three ways of using margin, padding etc...

```
p {  
    margin-top: 5px;  
    margin-right: 10px;  
    margin-bottom: 5px;  
    margin-left: 10px;  
}  
  
p {  
    /* shorthand version (top, right, bottom, left) */  
    margin: 5px 10px 5px 10px;  
}  
  
p {  
    /* shorthand version (top & bottom, right & left) */  
    margin: 5px 10px;  
}
```

Border properties

```
.header-card {  
    color: ■#fff;  
    background-color: □#333;  
  
    /* Border properties (size, color, type) */  
    border-top: 5px ■red solid;  
    border-right: 5px ■red solid;  
    border-bottom: 5px ■red solid;  
    border-left: 5px ■red solid;  
}
```



Shorthand - if all the sides are the same

```
.header-card {  
    color: black;  
    background-color: white;  
  
    /* Border properties (size, color, type) */  
    border: 5px solid black;  
}
```

Hello World!

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestiae iusto perferendis cum excepturi velit porro praesentium nostrum consequuntur sunt illo aliquam, hic odit repellat minus incident debitis laudantium enim nam quod labore id culpa paratur. Cupiditate, voluptatem. Consectetur, repellendus repudiandae.

NOTE: Within the class, the properties are set from top to bottom SO, if a property is set after the same property is set... the last one is used... for example:

Here we are setting ALL the border to 5px, green and solid. But, after that definition - we we change the border-bottom to red, then everything is set to green except the bottom is red...

```
.header-card {  
    color: ■#fff;  
    background-color: □#333;  
  
    /* Border properties (size, color, type) */  
    border: 5px □green solid;  
  
    border-bottom: 10px ▢red dotted;  
}
```

Hello World!

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestiae iusto perferendis cum excepturi velit porro praesentium nostrum consequuntur sunt illo aliquam, hic odit repellat minus incident debitis laudantium enim nam quod labore id culpa paratur. Cupiditate, voluptatem. Consectetur, repellendus repudiandae.

Targeting Selectors within styles

Sometimes, we want to modify an element style but only within a container (a div). Let's say, for example, we are displaying a list of blog posts and we want to change the <P> paragraph element style but not ALL paragraph element styles on the website. Let's modify and simplify our site : html

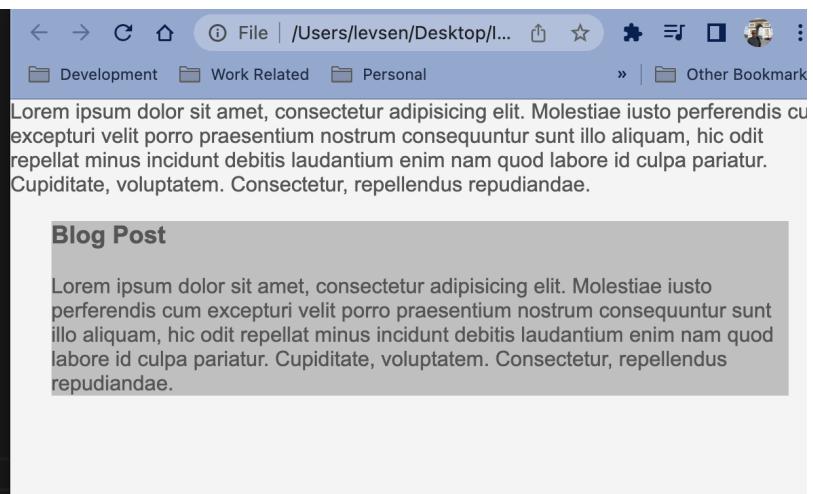
```
<body>

    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
       Molestiae iusto preferendis cum excepturi velit porro praesentiu
    </p>

    <div class="container">
        <div class="header-card">
            <h3>Blog Post</h3>
            <p>Lorem ipsum dolor <span>sit amet, consectetur </span>adipisci
               Molestiae iusto preferendis cum excepturi velit porro pra
            </p>
            <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.
               Molestiae iusto preferendis cum excepturi velit porro pra
            </p>
        </div>
    </div>

</body>
```

Css and output:



The screenshot shows a browser window with the following details:

- Left Panel (Code View):** Contains the CSS code for targeting specific paragraph styles within a container.
- Right Panel (Output View):** Shows the rendered HTML structure of a blog post. The first paragraph is styled with a margin of 5px 10px, padding of 0, and margin of 0. The second paragraph is contained within a div with class "header-card", which has a width of 90% and margin set to auto. The third paragraph is part of the "header-card" div and has a background color of #bfefef.

```
p {
    margin: 5px 10px;
    padding: 0;
    margin: 0;
}

.container {
    width: 90%;
    margin: auto;
}

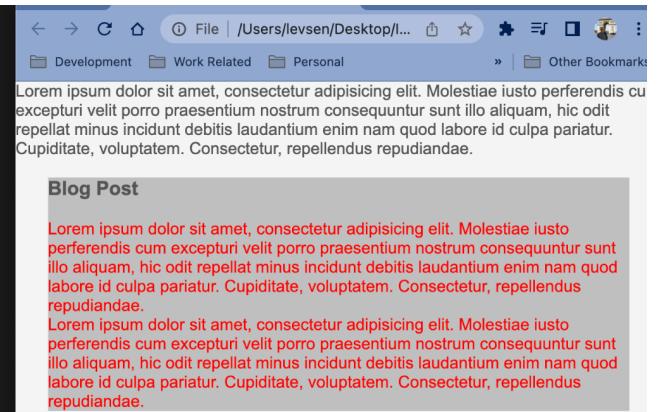
.header-card {
    background-color: #bfefef;
```

Targeting elements within other elements to change the style is as follows

```
.container p {
```

Changing the <p>

```
p {  
    margin: 5px 10px;  
    padding: 0;  
    margin: 0;  
}  
  
.container {  
    width: 90%;  
    margin: auto;  
}  
  
.container p {  
    color: red;  
}
```



We can even target spans inside of a paragraph BUT only when the paragraph is inside the container

```
.container p span {  
    color: green;  
}
```

Blog Post

Placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestiae iusto perferendis cum excepturi velit porro praesentium illo aliquam, hic odit repellat minus incident debitis laudantium enim nam quod labore id culpa pariatur. Cupiditate, voluptatem. Consectetur, repellendus repudiandae.' The word 'adipisicing' is colored green, demonstrating the targeting of a span element within a paragraph.

That's all for this overview... this can take days to go through all of the possibilities

Let's take a look at Lists.. But before, let's cleanup our html and css files - we are working towards putting together a blog listing site

```
<body>  
  
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.  
| Molestiae iusto perferendis cum excepturi velit porro praesentium  
</p>  
  
<div class="container">  
    <div class="header-card">  
        <h3>Blog Post One</h3>  
        <p>Lorem ipsum dolor adipisicing elit.  
        | Molestiae iusto perferendis cum excepturi velit porro praesentium  
        </p>  
    </div>  
    <div class="header-card">  
        <h3>Blog Post Two</h3>  
        <p>Lorem ipsum dolor adipisicing elit.  
        | Molestiae iusto perferendis cum excepturi velit porro praesentium  
        </p>  
    </div>  
</div>  
  
</body>
```

And the CSS cleanup and final output

The screenshot shows a browser window with a navigation bar at the top. Below the navigation bar, there are two blog post cards. Each card has a title and a snippet of text. The CSS on the left is used to style these elements.

```
p {  
    margin: 5px 10px;  
    padding: 0;  
    margin: 0;  
}  
  
.container {  
    width: 90%;  
    margin: auto;  
}  
  
.header-card {  
    background-color: #fbfbfb;  
}
```

Blog Post One

Blog Post Two

Development Work Related Personal > Other Bookmarks

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestiae iusto perferendis cum excepturi velit porro praesentium nostrum consequuntur sunt illo aliquam, hic odit repellat minus incident debitis laudantium enim nam quod labore id culpa pariatur. Cupiditate, voluptatem. Consectetur, repellendus repudiandae.

Lorem ipsum dolor adipisicing elit. Molestiae iusto perferendis cum excepturi velit porro praesentium nostrum consequuntur sunt illo aliquam, hic odit repellat minus incident debitis laudantium enim nam quod labore id culpa pariatur. Cupiditate, voluptatem. Consectetur, repellendus repudiandae.

Now, lets add a HTML UL list of categories. **NOTE: it's always a good idea to add class to the elements as you're typing them in... preplan. Then, you can style it by adding the classes.**

Html

```
<div class="category-container">  
    <h3>Categories</h3>  
    <ul>  
        <li><a href="#">Category 1</a></li>  
        <li><a href="#">Category 2</a></li>  
        <li><a href="#">Category 3</a></li>  
        <li><a href="#">Category 4</a></li>  
        <li><a href="#">Category 5</a></li>  
    </ul>  
</div>
```



CSS

```
.category-container {  
    background-color: #5353c9;  
}  
  
.category-container h3 {  
    text-align: center;  
    color: white;  
}  
  
.category-container ul{  
    list-style: none;  
    padding: 0;  
}  
.category-container ul li{  
    margin-top: 5px;  
    margin-bottom: 5px;  
}  
  
.category-container ul li a {  
    text-decoration: none;  
    color: white;  
}  
  
.category-container ul li a:hover {  
    color: red;  
}
```

*** INSTRUCTOR:

Walk through the CSS styling

Forms

First, create a new html file named Contactus.html and add the following

1. Create and basic structure (type: html:5 tab)
2. Create an empty CSS file named forms.css in the styles folder and create a link to it in the header section

```
<head>
  <title>Forms</title>
  <link rel="stylesheet" href="styles/forms.css">
</head>
```

3. Add the following to create a basic form

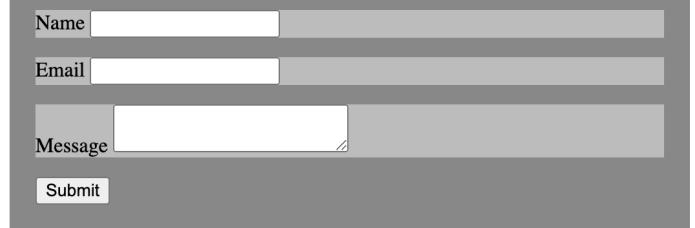
```
<form>
  <div class="form-group">
    <label>Name </label>
    <input type="text" name="name">
  </div>
  <div class="form-group">
    <label>Email </label>
    <input type="text" name="email">
  </div>
  <div class="form-group">
    <label>Message </label>
    <textarea type="text" name="message"></textarea>
  </div>
  <input class="button" type="submit" value="Submit">
</form>
```

4. View the page... should look as follows

The image shows a screenshot of a web browser displaying a contact form. The form is contained within a dark grey box. It includes three text input fields: one for 'Name' (with a placeholder 'John Doe'), one for 'Email' (with a placeholder 'john.doe@example.com'), and one for 'Message' (with a placeholder 'Type your message here...'). Below these fields is a single button labeled 'Submit'.

Let's add some CSS to our form... starting with adding a little spacing between the form items

```
form {  
    padding: 20px;  
    background-color: #888888;  
}  
  
form .form-group {  
    background-color: #bcbcbc;  
    margin-bottom: 15px;  
}
```

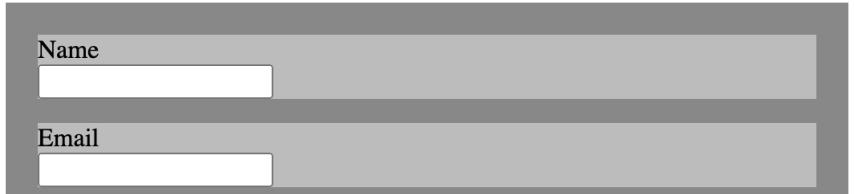


A screenshot of a web browser showing a form with three input fields: Name, Email, and Message, each in its own grey box. Below the inputs is a Submit button.

Yes, it's super ugly but it shows the different areas better so we can see the effects of our CSS changes

Next, let's move the labels to their own line

```
form label {  
    display: block;  
}
```

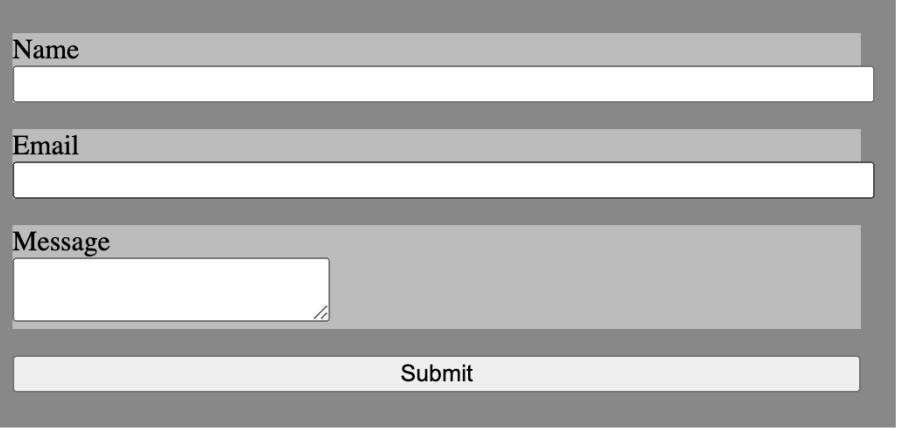


A screenshot of a web browser showing two input fields: Name and Email. Each has a label above it ('Name' and 'Email') on a new line. The input fields are in separate grey boxes.

*** remember, block places things on their own line

Next, we want to extend the inputs so that it uses the full width on the div that surrounds them

```
form input {  
    width: 100%;  
}
```



A screenshot of a web browser showing three input fields: Name, Email, and Message. Each has a label above it ('Name', 'Email', 'Message') on a new line. The input fields are now fully extended to fit the width of their parent divs, which are grey boxes.

**** problem : changing the 'input' to be 100% isn't going to work because buttons, radio buttons, etc are all inputs. Also, the TextArea isn't extended but it's not an input

We can fix this by targeting certain types of inputs - therefore, allowing buttons (for instance) to appear properly)

```
form input[type="text"], form textarea {  
    width: 100%;  
}
```

A screenshot of a web browser showing a contact form. The form consists of three text input fields: 'Name', 'Email', and 'Message', each with a placeholder text area below it. Below the text areas is a single 'Submit' button.

Let's add some styling to the button

```
.button {  
    background-color: #333;  
    color: white;  
    padding: 10px 15px;  
    border: none;  
}  
  
.button:hover {  
    background-color: #444;  
}
```

A screenshot of a web browser showing the same contact form as before, but now the 'Submit' button is styled with a dark gray background and white text, matching the styling of the other form inputs.

Alignment and Floating

Let's add three blocks of text into our forms.html file

```
<!-- Alignment and floating -->
<div class="sample-block">
  <h3>Block One</h3>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.</p>
</div>
<div class="sample-block">
  <h3>Block Two</h3>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.</p>
</div>
<div class="sample-block">
  <h3>Block Three</h3>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.</p>
</div>
```

Block One

Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.

Block Two

Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.

Block Three

Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.

Let's say we want them side-by-side each other - on the same row we can simple add the following class - the 'sample-block' class

```
.sample-block {
  float: left;
  width: 33.3%
}
```

Now what we get is the following

Block One

Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.

Block Two

Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.

Block Three

Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.

Using floats can cause some strange behaviour. Let's take a look at a couple issues that can occur and how to fix them.

To demonstrate this, we can add a couple of extra DIVs with some paragraph text and also add a border, padding or a margin on the three boxes we do the following.

1. We will add a couple DIVs into the html - below the boxes

```
<div>
|   Lorem, ipsum dolor sit amet consectetur adipisicing elit
</div>
<div>
|   Lorem, ipsum dolor sit amet consectetur adipisicing elit
</div>
```

2. Add a border with some padding

```
.sample-block {
    float: left;
    border: 1px solid #333;
    padding: 15px;
    width: 33.3%
}
```

Doing this, we now get the following output

Block One Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.	Block Two Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.	Lorem, ipsum dolor sit amet consectetur adipisicing elit. Dolor atque ut nulla obcaecati quisquam. Est accusantium repudiandae necessitatibus unde aliquid! Lorem, ipsum dolor sit amet consectetur adipisicing elit. Dolor atque ut nulla obcaecati quisquam. Est accusantium repudiandae necessitatibus unde aliquid!
Block Three Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.		

So, to fix the first problem, we need to place a new DIV after our float DIV's which basically force any further markup to start below any of the previous FLOATS, as follows

```
<div class="clear-float"></div>
```

In the CSS, the clear-float class is simply

```
.clear-float {
    clear: both;
}
```

Now, the output should look as follows

Block One Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.	Block Two Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.
Block Three Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.	
Lorem, ipsum dolor sit amet consectetur adipisicing elit. Dolor atque ut nulla obcaecati quisquam. Est accusantium repudiandae necessitatibus unde aliquid! Lorem, ipsum dolor sit amet consectetur adipisicing elit. Dolor atque ut nulla obcaecati quisquam. Est accusantium repudiandae necessitatibus unde aliquid!	

The other problem we had was that the three boxes don't fit across the page because we are setting each box to be 33.3% of the page width. And, when you add the border of 1px and then the padding of 15px - this becomes wider than 33.3% of the page... so two of them fit - but, the last one doesn't.

So to fix this, we can tell the CSS class the account for the padding, margin, and border sizes in the 33.3% assigned in the width property - like this

```
.sample-block {  
    float: left;  
    border: 1px solid #333;  
    padding: 15px;  
    width: 33.3%;  
    box-sizing: border-box;  
}
```

Block One Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.	Block Two Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.	Block Three Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam non fugit ipsum, qui animi necessitatibus.
Lore, ipsum dolor sit amet consectetur adipisicing elit. Dolor atque ut nulla obcaecati quisquam. Est accusantium repudiandae necessitatibus unde aliquid! Lore, ipsum dolor sit amet consectetur adipisicing elit. Dolor atque ut nulla obcaecati quisquam. Est accusantium repudiandae necessitatibus unde aliquid!		

Assignment

Using the following HTML use CSS to make it appear as the final image below

```
assignment.html > html > body > header#main-header
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>Assignment</title>
5      <link rel="stylesheet" href="styles/assignment-styles.css">
6  </head>
7  <body>
8      <!-- Header/Logo Bar -->
9      <header id="main-header">
10         <div class="container">
11             | <h1>My Website</h1>
12         </div>
13     </header>
14
15     <!-- Navigation -->
16     <nav id="navbar">
17         <div class="container">
18             <ul>
19                 | <li><a href="#">Home</a></li>
20                 | <li><a href="#">Products</a></li>
21                 | <li><a href="#">Services</a></li>
22                 | <li><a href="#">Contact Us</a></li>
23             </ul>
24         </div>
25     </nav>
26
27     <div class="container">
28
29         <!-- Showcase area -->
30         <section id="showcase">
31             <div class="container">
32                 | <h1>Lorem ipsum dolor, sit amet consectetur cumque cupiditate beatae
33             </div>
34         </section>
35
36         <!-- main area -->
37         <section id="main">
38             <h2>Welcome</h2>
39             | <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Id dolore
40         </section>
41
42         <!-- sidebar area -->
43         <aside id="sidebar">
44             | <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Commodi, cumque
45         </aside>
46
47     </div>
48
49     <footer id="footer">Copyright &copy;2023 My Website</footer>
50
51 </body>
52 </html>
```



Using ONLY CSS, you should be able to reproduce (more or less) the site below.

Note: the image can be anything wish - this is just a sample from google images doing a search for 'abstract art'

My Website

Home Products Services Contact Us



**Lore ipsum dolor, sit amet consectetur
cumque cupiditate beatae !**

Welcome

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Id dolore commodi nemo doloribus excepturi sint fugit. Sed unde praesentium a. Voluptas quos quas non magnam ducimus aliquid! Nulla corrupti cumque eveniet iusto labore. Est eos libero doloribus aliquam reiciendis maiores tempora illo temporibus ad, eveniet, esse eius reprehenderit dolor. Ducimus!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Commodi, cumque praesentium libero cupiditate beatae voluptatem.

Copyright ©2023 My Website

Additional notes about responsive websites

In our assignment, we have a sidebar with text beside the main content. But, when the width narrow - like on a smart phone... the text gets squished. As shown on the right.

In CSS, we can do a ‘media query’ and apply different css when - for instance - a width is below a certain value. Here is the CSS

```
@media(max-width: 600px) {  
  
    #main {  
        width: 100%;  
        float: none;  
    }  
  
    #sidebar {  
        width: 100%;  
        float: none;  
    }  
}
```

When the width is 600px (or less), apply the following changes to the main and sidebar style properties. The end result looks like the image on the right.

Removing the float and setting the width to 100% makes it appear stacked on top of each other.

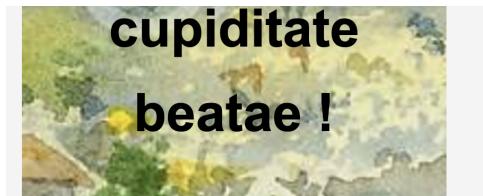
This is how a ‘Responsive Website’ should react to varying devices.



Welcome

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Id dolore commodi nemo doloribus excepturi sint fugit. Sed unde praesentium a. Voluptas quos quas non magnam ducimus aliquid! Nulla corrupti cumque eveniet iusto labore. Est eos libero doloribus aliquam reiciendis maiores tempora illo temporibus ad, eveniet, esse eius reprehenderit dolor. Ducimus!

beatae !
Welcome
Lorem ipsum
dolor sit amet
consectetur
adipisicing
elit.
Commodi,
cumque
praesentium
libero
cupiditate
beatae
voluptatem.



Welcome

Cupiditate
beatae !
Welcome
Lorem ipsum dolor sit amet consectetur, adipisicing elit. Id dolore commodi nemo doloribus excepturi sint fugit. Sed unde praesentium a. Voluptas quos quas non magnam ducimus aliquid! Nulla corrupti cumque eveniet iusto labore. Est eos libero doloribus aliquam reiciendis maiores tempora illo temporibus ad, eveniet, esse eius reprehenderit dolor. Ducimus!

cupiditate
beatae !
Welcome
Lorem ipsum dolor sit amet consectetur adipisicing elit. Commodi, cumque praesentium libero cupiditate beatae voluptatem.