Millersville University

Lombardo College of Business

Department of Management and Marketing

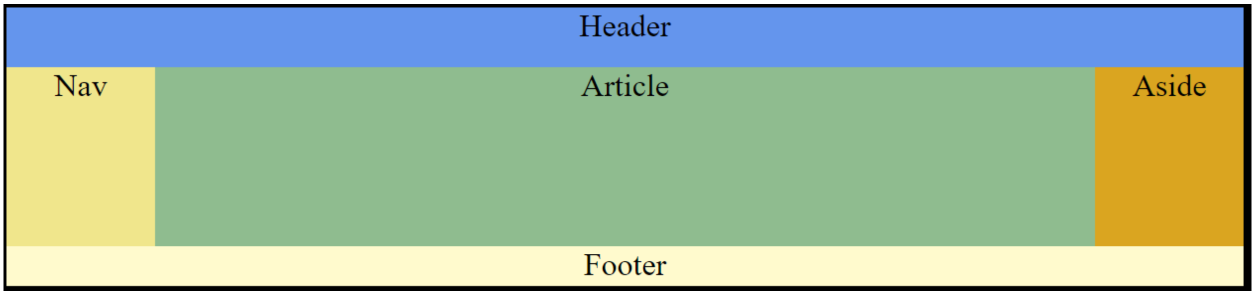
CSS Lab – Part 6 – Tailwind Grid Areas (2025)

In this lab, we will be learning about CSS Grid Areas and how to implement arbitrary values in Tailwind CSS. First create a new folder in your CSS folder called Lab 6. Start a new file called css-grid-area.html.

Enter the following code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Grid Area Example</title>
    <style>
      body {
        background: black;
        font-size: 2em;
        text-align: center;
        display: grid;
        grid-template-columns: 150px auto 150px;
        grid-template-rows: 60px 180px 40px;
        grid-template-areas:
          "head head head"
          "nav article aside"
          "foot foot foot";
      }

      header {
        background: cornflowerblue;
        grid-area: head;
      }

      nav {
        background: khaki;
        grid-area: nav;
      }
      article {
        background: darkseagreen;
        grid-area: article;
      }

      aside {
        background: goldenrod;
        grid-area: aside;
      }

      footer {
        background: lemonchiffon;
        grid-area: foot;
      }
    </style>
  </head>
  <body>
    <header>Header</header>
    <nav>Nav</nav>
    <article>Article</article>
    <aside>Aside</aside>
    <footer>Footer</footer>
  </body>
</html>
```

Run the code and you will get the following:



Assignment:

The code that was written was in plain CSS. Your task is to rewrite the code using Tailwind CSS. Create a new file called tailwind-grid-area.html and use the Emmet macro !(enter) to have a starting html file.

Add in the header a script tag to load Tailwind CSS v4.

```
<script src="https://unpkg.com/@tailwindcss/browser@4"></script>
```

You will want to make sure you have the Tailwind CSS Visual Studio Code extension installed and a tailwind.config.ts file in the same folder for IntelliSense to work. Next, place the following from the first file into the body:

```
    <header>Header</header>
    <nav>Nav</nav>
    <article>Article</article>
    <aside>Aside</aside>
    <footer>Footer</footer>
```

Next your task is to go through each tag, the body, header, nav, article, aside, and footer and insert the valid tailwind code that results in code we had in the first file.

You should have <body class= “enter all your tailwind classes here”> for each tag. You will not be placing anything in the CSS except for a file called tailwindbase.css with the following:

```
@tailwind base; /* Preflight will be injected here */
@tailwind components;
@tailwind utilities;
```

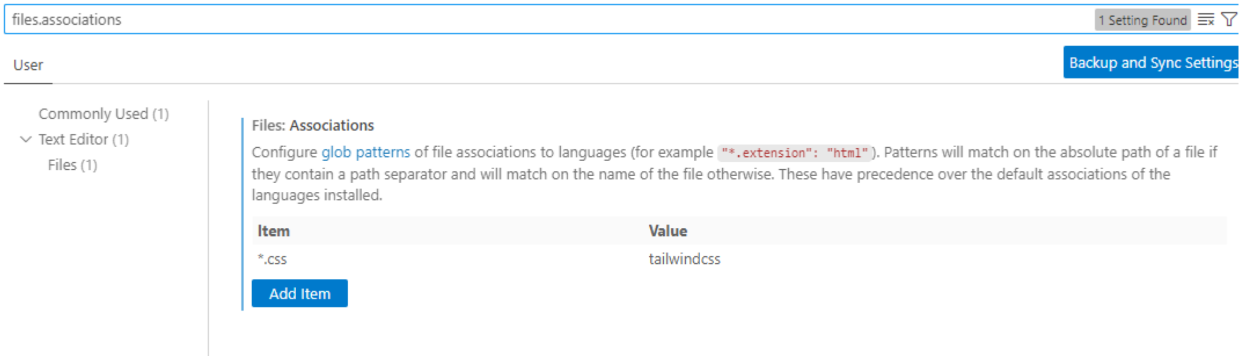
If you are getting yellow warnings for the @tailwind, make sure to finish setting up the Tailwind CSS extension by searching in your settings for:

files.associations

Use the files.associations setting to tell VS Code to always open .css files in Tailwind CSS mode:

```
"files.associations": {
  "*.css": "tailwindcss"
}
```

Your settings should look like the following:

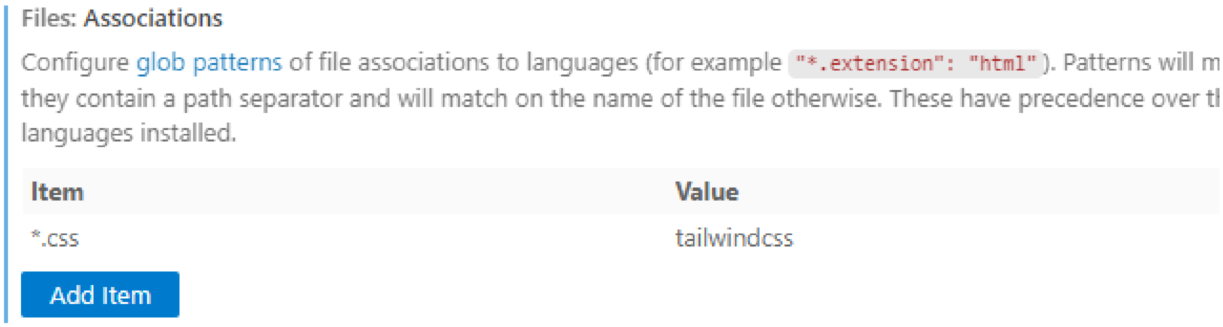


files.associations

Files: Associations

Configure glob patterns of file associations to languages (for example `"*.extension": "html"`). Patterns will match on the absolute path of a file if they contain a path separator and will match on the name of the file otherwise. These have precedence over the default associations of the languages installed.

Item	Value
*.css	tailwindcss



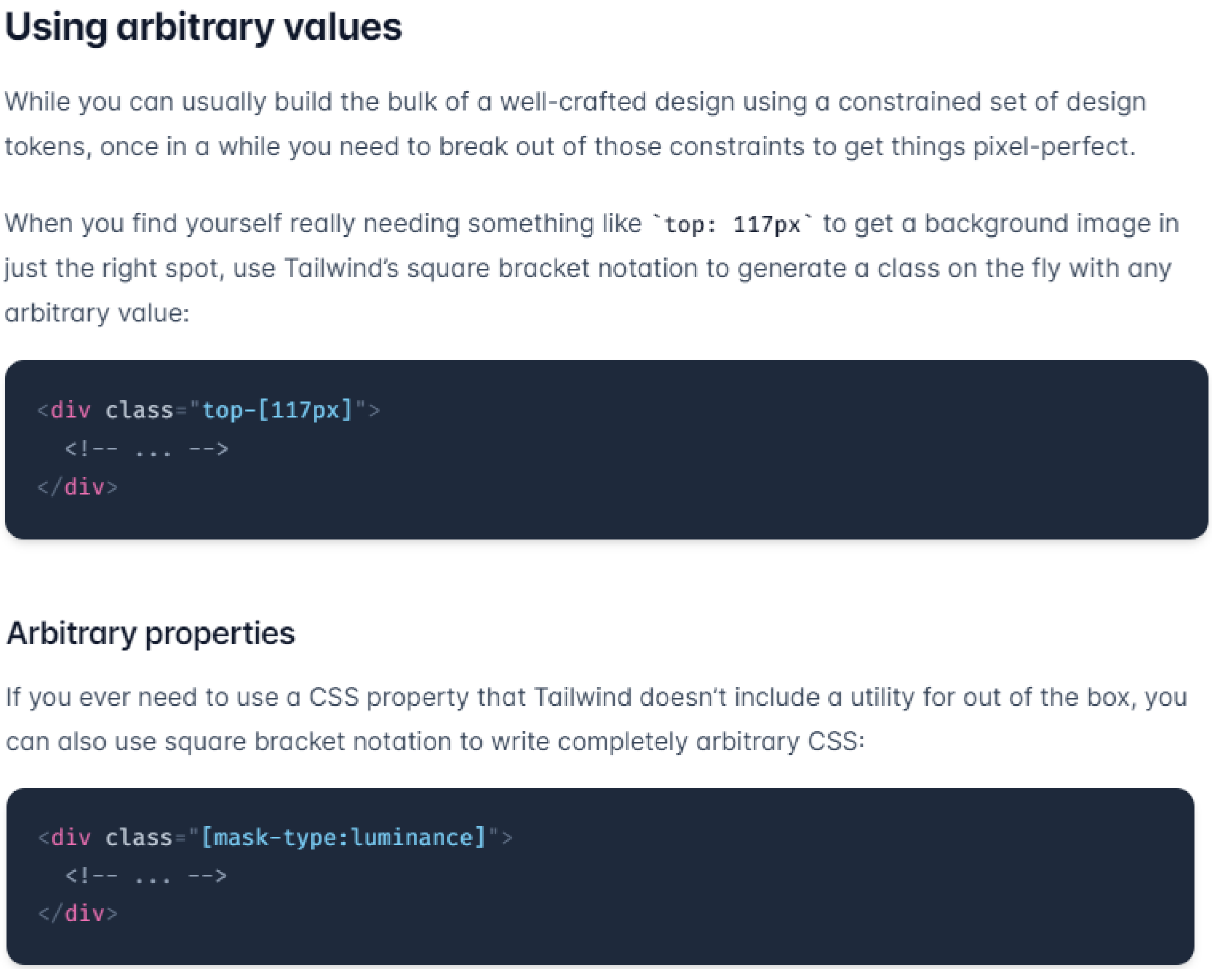
Files: **Associations**

Configure glob patterns of file associations to languages (for example `"*.extension": "html"`). Patterns will m they contain a path separator and will match on the name of the file otherwise. These have precedence over tl languages installed.

Item	Value
*.css	tailwindcss

Link this CSS file as normal in your head tag. It is suggested you use the documentation to what tailwind commands are equivalent to plain CSS.

Sometimes you will find a CSS property you need, and it is not in Tailwind CSS yet. Take for example grid-template-areas. The project will not work without adding in a grid-template-areas but how to do it? The answer is in the documentation under add custom styles: https://tailwindcss.com/docs/adding-custom-styles#using-arbitrary-values



Using arbitrary values

While you can usually build the bulk of a well-crafted design using a constrained set of design tokens, once in a while you need to break out of those constraints to get things pixel-perfect.

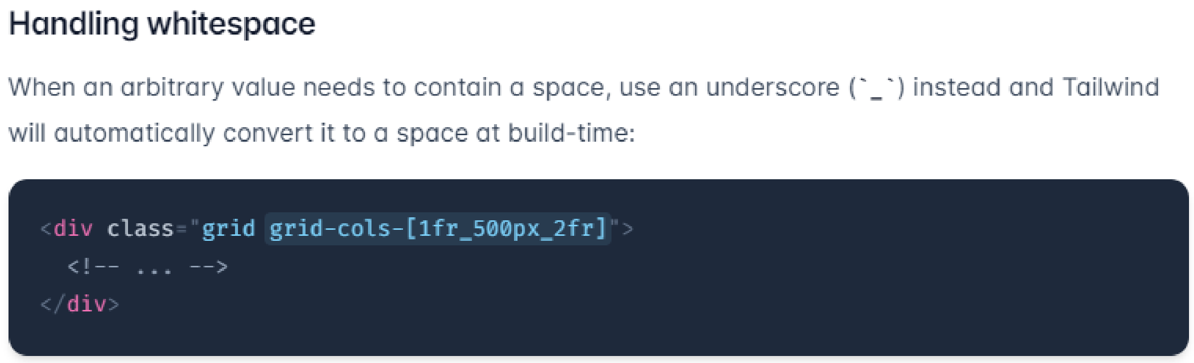
When you find yourself really needing something like `top: 117px` to get a background image in just the right spot, use Tailwind's square bracket notation to generate a class on the fly with any arbitrary value:

```html
<div class="top-[117px]">
  <!-- ... -->
</div>
```

Arbitrary properties

If you ever need to use a CSS property that Tailwind doesn't include a utility for out of the box, you can also use square bracket notation to write completely arbitrary CSS:

```html
<div class="[mask-type:luminance]">
  <!-- ... -->
</div>
```



Handling whitespace

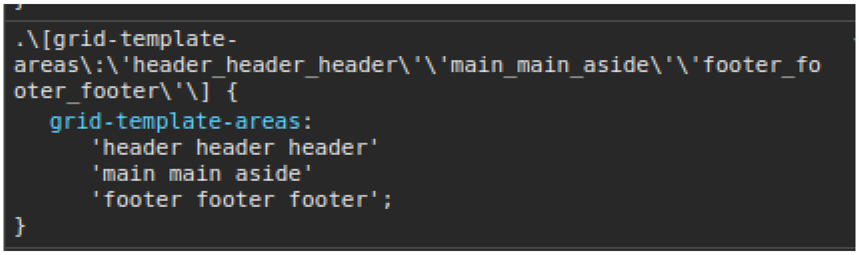
When an arbitrary value needs to contain a space, use an underscore (`_`) instead and Tailwind will automatically convert it to a space at build-time:

```
<div class="grid grid-cols-[1fr_500px_2fr]">
  <!-- ... -->
</div>
```

Using these three ideas you can do something like:

```
[grid-template-areas:'header_header_header''main_main_aside''footer_footer_footer']
```

To generate css:



```
.\[grid-template-
areas\:\'header_header_header\'\'main_main_aside\'\'footer_fo
oter_footer\'\] {
   grid-template-areas:
      'header header header'
      'main main aside'
      'footer footer footer';
}
```

Complete the project and zip your folder containing all your files to CSS Lab 6.zip and upload to the drop box. Send your CSS Lab 6 to GitHub.

After you submit your project, you can look at a tailwind plug-in that adds grid-template-areas to Tailwind CSS. You would run the command:

```
npm install --save-dev @savvywombat/tailwindcss-grid-areas
```

You will need to be running a node.js project though for this to work and this will be covered later in the course. See https://savvywombat.com.au/tailwind-css/grid-areas for more information. If you are interested in grid-areas and want to use the plugin later in the course, come back the this statement when we have node.js installed.