

CSS Grid Layout

CSS3



What is CSS Grid Layout?

[CSS Grid Layout](#) affords us a powerful way to layout our web pages. In using “grid” we are able to decide how many rows and columns elements can take up. Note in that this method of layout includes both rows and columns, it is considered a two-dimensional layout system.

A core difference between CSS Flexible Box Layout and CSS Grid Layout is that fundamentally:

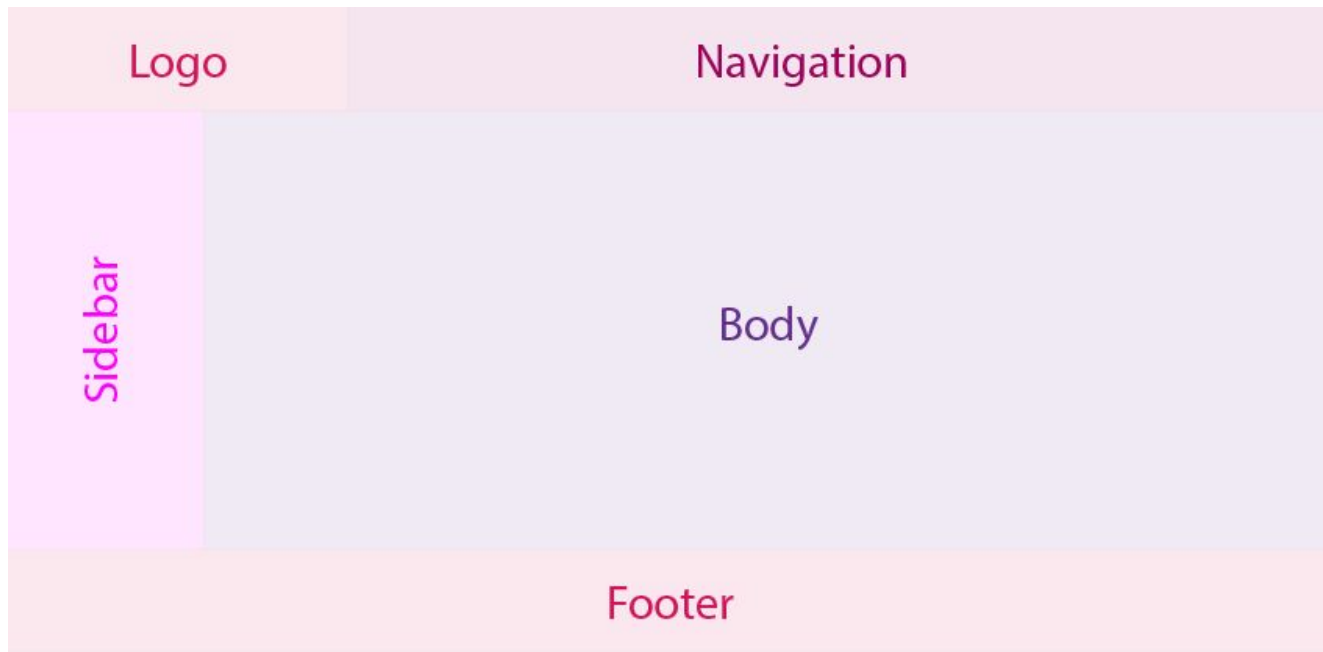
1. CSS Flexible Box Layout is used to organize and manipulate elements into rows **or** columns.
2. CSS Grid Layout is used to organize and manipulate elements within rows **and** columns.

Before you really get into writing code for grids, it can be helpful to get in the habit of breaking down your desired layout into potential rows and columns. Sketching and planning, even in small amounts, will go a long way in terms of giving you an idea on how to approach the layout from a CSS perspective.

Let’s examine a fairly common layout as a sketch or [wireframe](#), break it down into its rows and columns, and ultimately: program it.

How many rows and columns do you think the following would require?

Examining a Layout

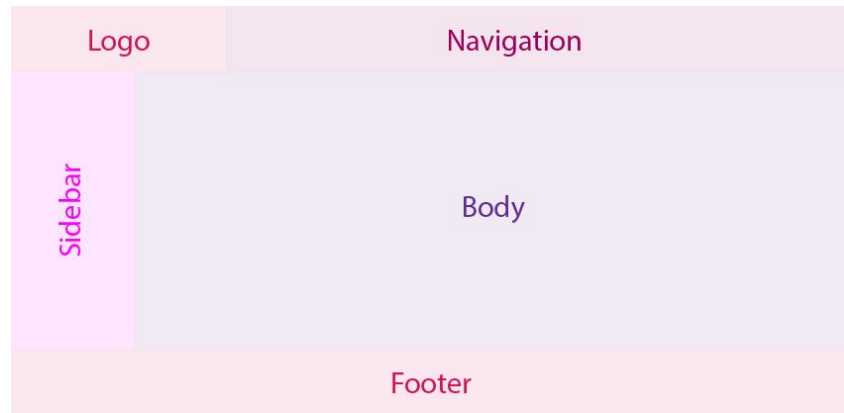


Breaking Down a Layout's Columns

Firstly, it is important to make a note of how many columns you may need.

Look carefully at any vertical lines in the design.
Draw a line for each, going from the top to the very bottom of the design.

How many columns do you see on the right?
Remember: columns are vertical sections.



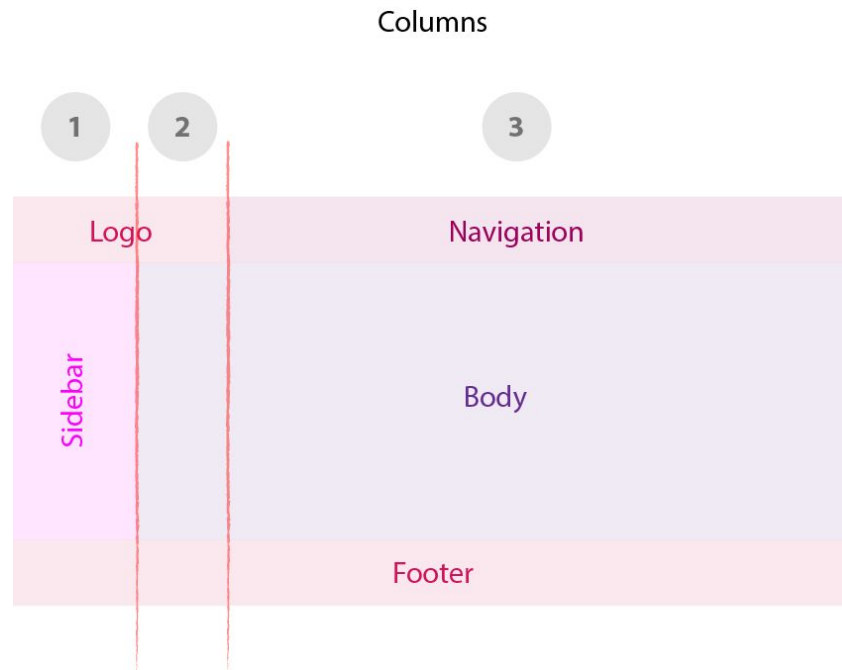
Breaking Down a Layout's Columns

Firstly, it is important to make a note of how many columns you may need.

Look carefully at any vertical lines in the design.
Draw a line for each, going from the top to the very bottom of the design.

How many columns do you see on the right?
Remember: columns are vertical sections.

There are three vertical slices in the simple design we are looking at!

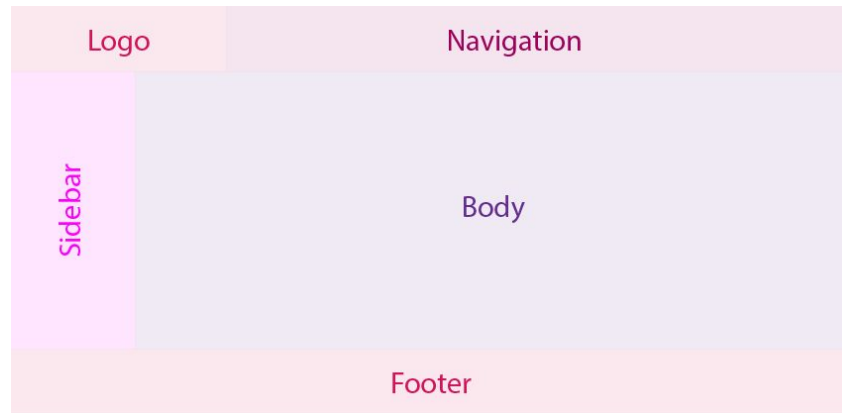


Breaking Down a Layout's Rows

Once you've assessed a design's columns, you should also make note of its rows—that is to say the horizontal slices within the design.

Look for any horizontal lines, and see about drawing a line from left to right across the entire image.

How many rows does this design feature?
Remember: rows are horizontal sections.



Breaking Down a Layout's Rows

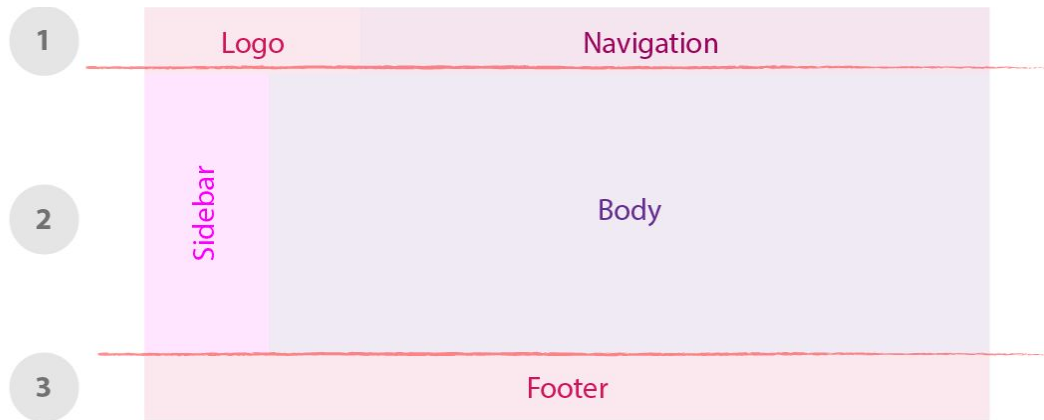
Once you've assessed a design's columns, you should also make note of its rows—that is to say the horizontal slices within the design.

Look for any horizontal lines, and see about drawing a line from left to right across the entire image.

How many rows does this design feature?
Remember: rows are horizontal sections.

There are three horizontal slices in the simple design we are looking at!

Rows



Programming a CSS Grid

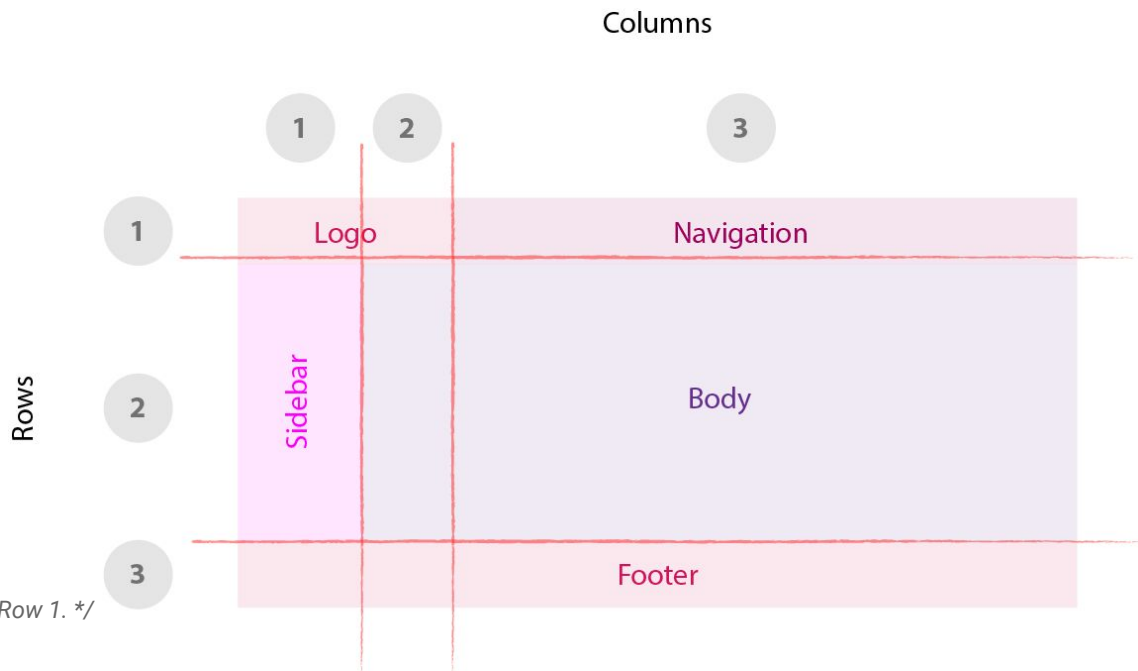
Try out the following...

HTML in the <body> of the web page:

```
<header>Logo</header>
<nav>Navigation</nav>
<aside>Sidebar</aside>
<article>Body</article>
<footer>Footer</footer>
```

CSS in the stylesheet for the web page:

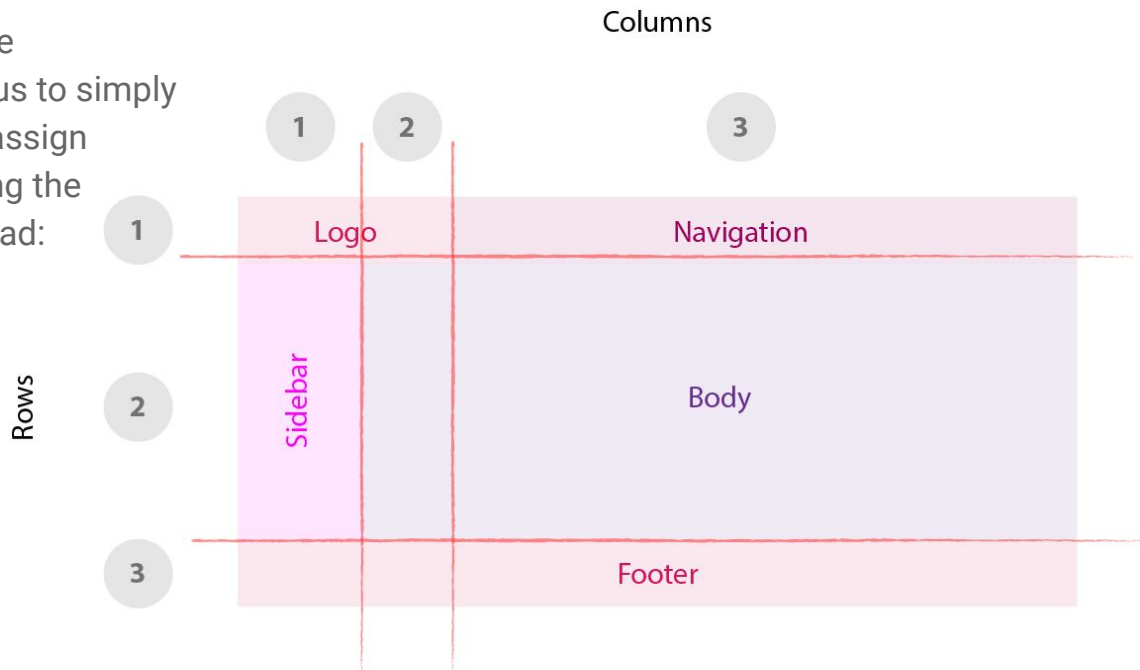
```
body {
  display: grid;
  grid-template-columns: repeat( 3, 1fr );
  gap: 0;
  grid-auto-rows: minmax( 100px, auto );
}
header { grid-column: 1 / span 2; grid-row: 1; } /* Column 1 and 2; Row 1. */
nav { grid-column: 3; grid-row: 1; } /* Column 3; Row 1. */
aside { grid-column: 1; grid-row: 2; } /* Column 1; Row 2. */
article { grid-column: 2 / span 2; grid-row: 2; } /* Column 1 and 2; Row 2. */
footer { grid-column: 1 / span 3; grid-row: 3; } /* Column 1 through 3; Row 3. */
```



Grid Template

CSS also affords us the more human-readable [grid-template](#) property. Grid template allows us to simply name areas in our grid, so that we may later assign elements to those locations. Let's try re-writing the previous experiment using this property instead:

```
body {  
  display: grid;  
  width: 100%;  
  height: auto;  
  grid-template: "logo logo navigation"  
                 "sidebar body body"  
                 "footer footer footer";  
}  
  
header { grid-area: logo; }  
nav { grid-area: navigation; }  
aside { grid-area: sidebar; }  
article { grid-area: body; }  
footer { grid-area: footer; }
```



Exercise

Sketch out another basic web design, see if you can add a bit of complexity.

Once the basic sketch or [wireframe](#) is complete, try to break it into rows and columns as we have in class.

Now that you have an idea of how the page is broken up, try to reproduce that layout using CSS code.

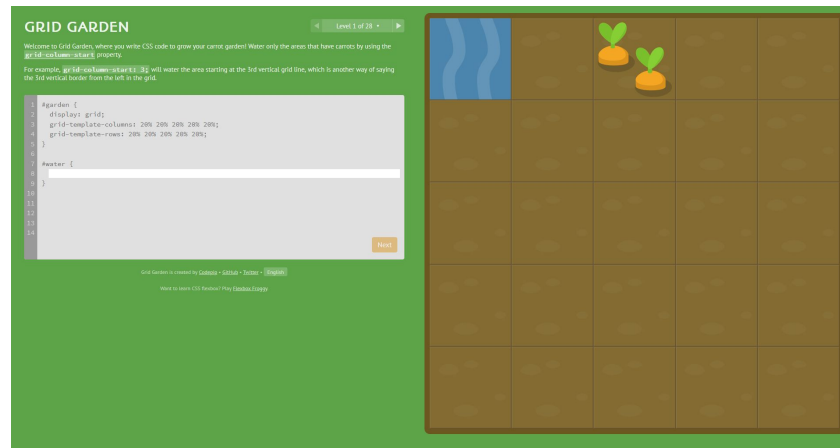
Practice Makes Perfect

Some of the best resources that exist on CSS Grid Layout to-date include:

- [MDN's CSS Grid Layout](#)
- [CSS Tricks' Getting Started with CSS Grid](#)
- [CSS Tricks' CSS Grid Starter Layouts](#)
- [CSS Tricks' A Complete Guide to Grid](#)
- [W3Schools' CSS Grid Layout Module](#)

There even exists an exceptional interactive practice resource:

- [Codepip's Grid Garden](#)



Recommended Reading

For more on how CSS grid layout operates:

- [Meyer, E. A; Weyl, E. \(October 2017\). CSS: The Definitive Guide, 4th Edition. O'Reilly Media, Inc.](#)
 - [Chapter 13. Grid Layout](#)