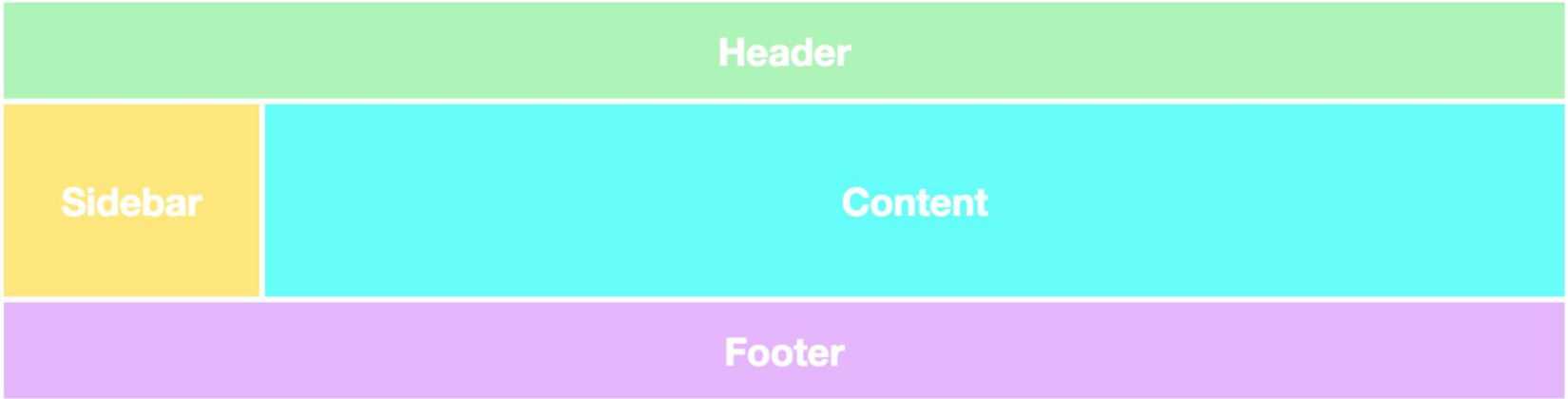
GRIDS

Grid layouts



Grid layouts are fundamental to the design of websites, and the CSS Grid module is the most powerful and easiest tool for creating it

Example Files

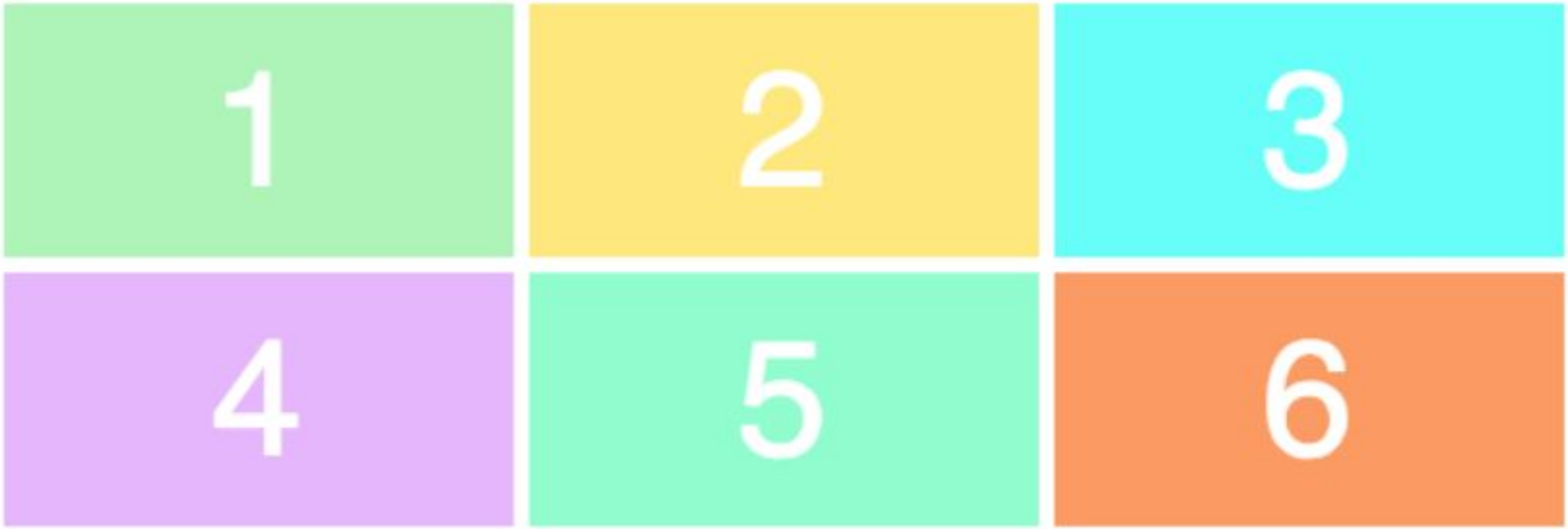
class5/floats_grid/grid.html

class5/floats_grid/grid_style/grid_style.css

Columns & Rows - 2 Dimensions

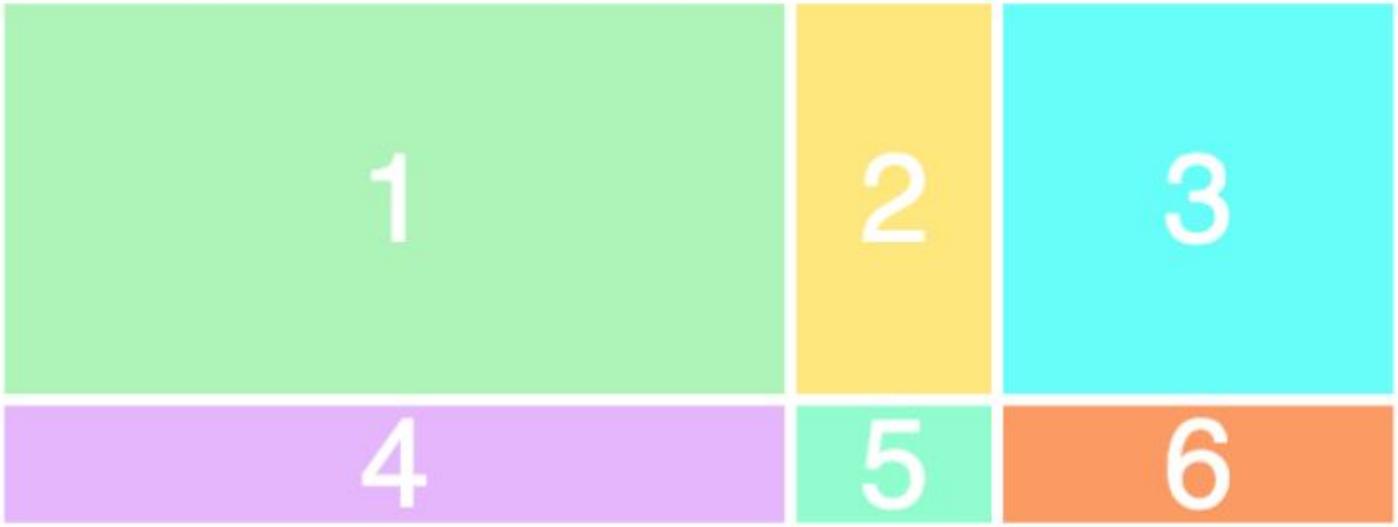
Let’s change the display in the CSS file to grid and define columns and rows (similar to an .html table):

```
.wrapper {
    display: grid;
    grid-template-columns: 100px 100px 100px;
    grid-template-rows: 50px 50px;
}
```

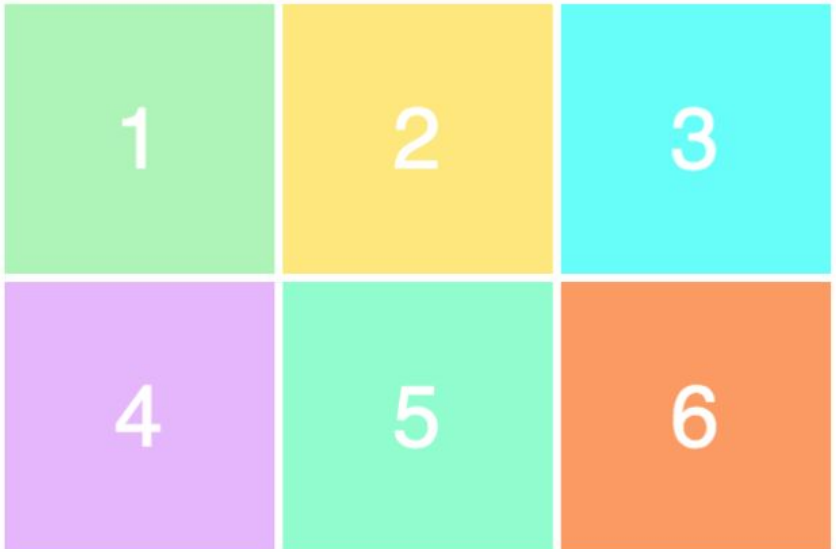


As we've written three values for grid-template-columns, we'll get three columns. We'll get two rows, as we've specified two values for the grid-template-rows.

The values dictate how wide we want our columns to be (100px) and how tall we'd want our rows to be (50px).




```
.wrapper {
    display: grid;
    grid-template-columns: 200px 50px 100px;
    grid-template-rows: 100px 30px;
}
```



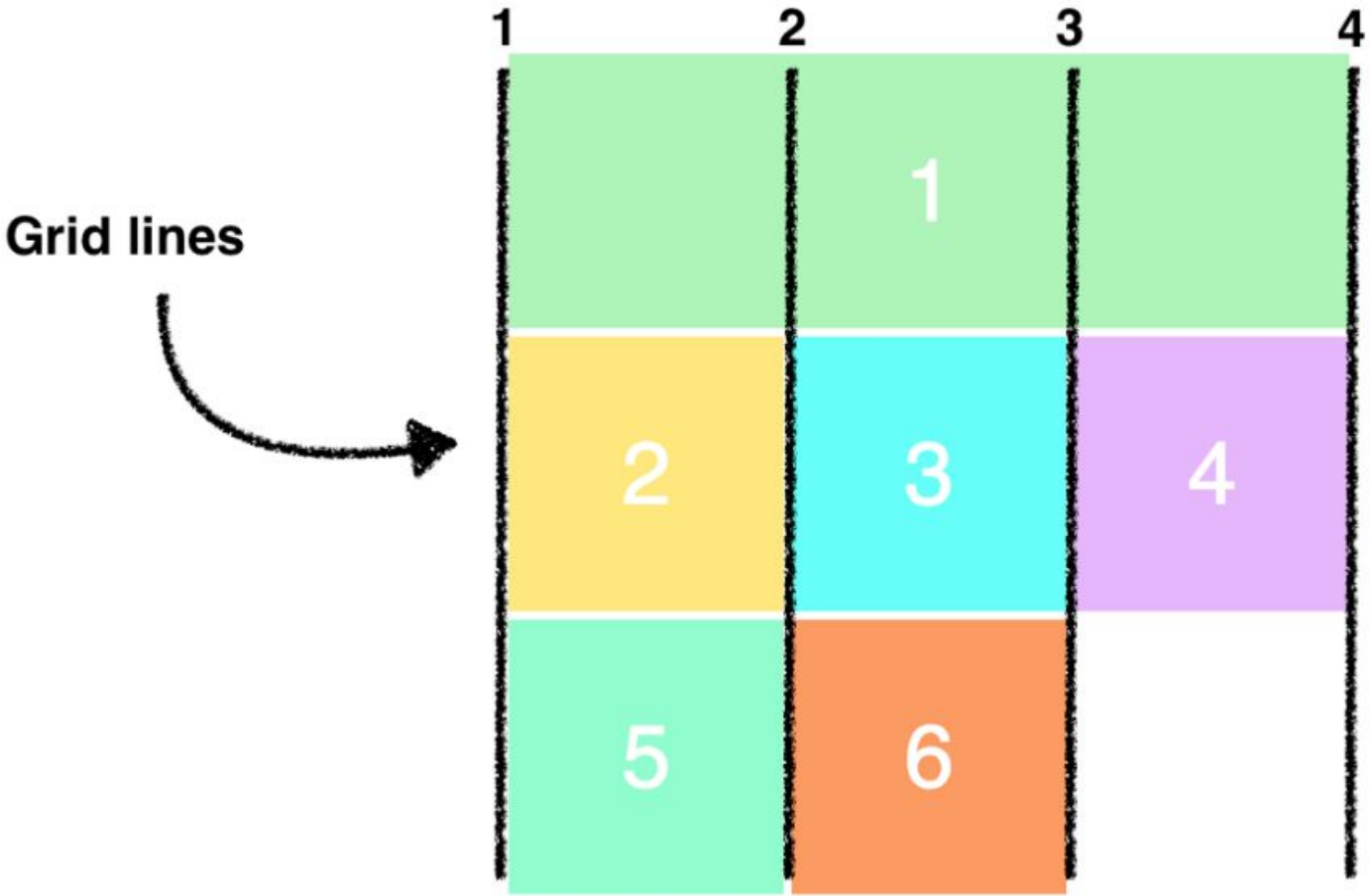

```
.wrapper {
    display: grid;
    grid-template-columns: 100px 100px 100px;
    grid-template-rows: 100px 100px 100px;
}
```

position & size

To position and resize the items we'll target them and use the grid-column and grid-row properties:

```
.item1 {
    grid-column-start: 1;
    grid-column-end: 4;
}
```

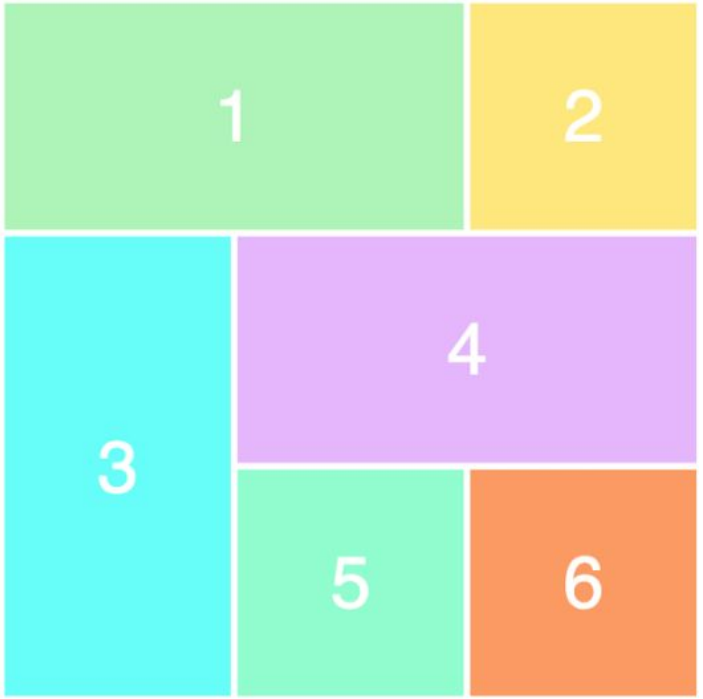
What we're saying here is that we want item1 to start on the first grid line and end on the fourth column line. In other words, it'll take up the entire row.


Grid lines
1
2
3
4
1
2
3
4
5
6

```
.item1 {
    grid-column-start: 1;
    grid-column-end: 3;
}

.item3 {
    grid-row-start: 2;
    grid-row-end: 4;
}

.item4 {
    grid-column-start: 2;
    grid-column-end: 4;
}
```



Static vs. Fluid Layouts

Static

A static page layout (sometimes called a “fixed” layout or “fixed width” layout) uses a preset page size and does not change based on the browser width. In other words, the page layout might have a permanent width of 960 pixels no matter what. This is how web pages were traditionally built for many years until modern influences like media queries and responsive design were introduced around the start of the 2010s.

Different devices will treat a static page layout in various ways, so the rendered page could be slightly unpredictable. For example, on a desktop browser, if the window is too small horizontally, then the page will be cut off and horizontal scroll bars will be displayed. On a mobile device like an iPhone, the page will be scaled automatically, allowing the user to zoom in on points of interest (provided that no metatags override this default behavior).

When new websites are built, most of them don’t opt for a static layout, because it means that the mobile experience will require a separate website. There are major exceptions, such as the online Apple.com store, but Apple is a unique case because a selling point of their mobile devices is that they can view static layouts. In other words, Apple doesn’t seem to be adopting responsive design just yet.

Static Example File

class5/layouts/static-page-layout-example

Fluid

A liquid page layout (sometimes called “fluid” or “fluid width”) uses relative units instead of fixed units. Typically a liquid layout will use percentages instead of pixels, but any relative unit of measurement will work, such as ems.

A liquid layout often will fill the width of the page, no matter what the width of the browser might be. Liquid layouts don’t require as much thought as a responsive design, but there are some major drawbacks at very large or very small browser widths. If the browser is very wide, some content might be stretched too far. On large screens, a single paragraph might run across the page on a single line. Conversely, a multi-column layout on a small screen could be too crowded for the content.

Fluid Example File

class5/layouts/liquid-page-layout-example