

Class 8: Page Layout

Grid-based layouts

Grids used in the wild

Layout templates

Grid systems and bootstraps

Page layout using floats

The clearfix hack and a modern alternative (display: flow-root;)

Content order

Page layout with CSS Grid

Fluid layouts with percentages

Fluid grid layouts with fr units

CSS Layout Challenge:

<https://www.websitearchitecture.co.uk/resources/design-for-web-content/examples/class-08-layout-workshop.zip>

References

Learning Web Design (5th Edition) by Jennifer Robbins

Head First HTML and CSS by Elizabeth Robson & Eric Freeman

HTML & CSS by Jon Duckett

Handcrafted CSS by Dan Cederholm

Above The Fold by Brian Miller

A Practical Guide to Designing for the Web by Mark Boulton

Ordering Disorder by Khoi Vinh

<https://read.compassofdesign.com/guides-gutters-and-grids-2ce6092fc3de>

<https://developer.mozilla.org/en-US/docs/Web/CSS/float>

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Floats

<https://css-tricks.com/all-about-floats/>

<https://css-tricks.com/snippets/css/clear-fix/>

<https://alistapart.com/article/fluidgrids/>

https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Introduction

<https://www.youtube.com/watch?v=FErRpy9Xfes>

<https://developer.mozilla.org/en-US/docs/Web/CSS/grid>

<https://developer.mozilla.org/en-US/docs/Web/CSS/grid-template-areas>

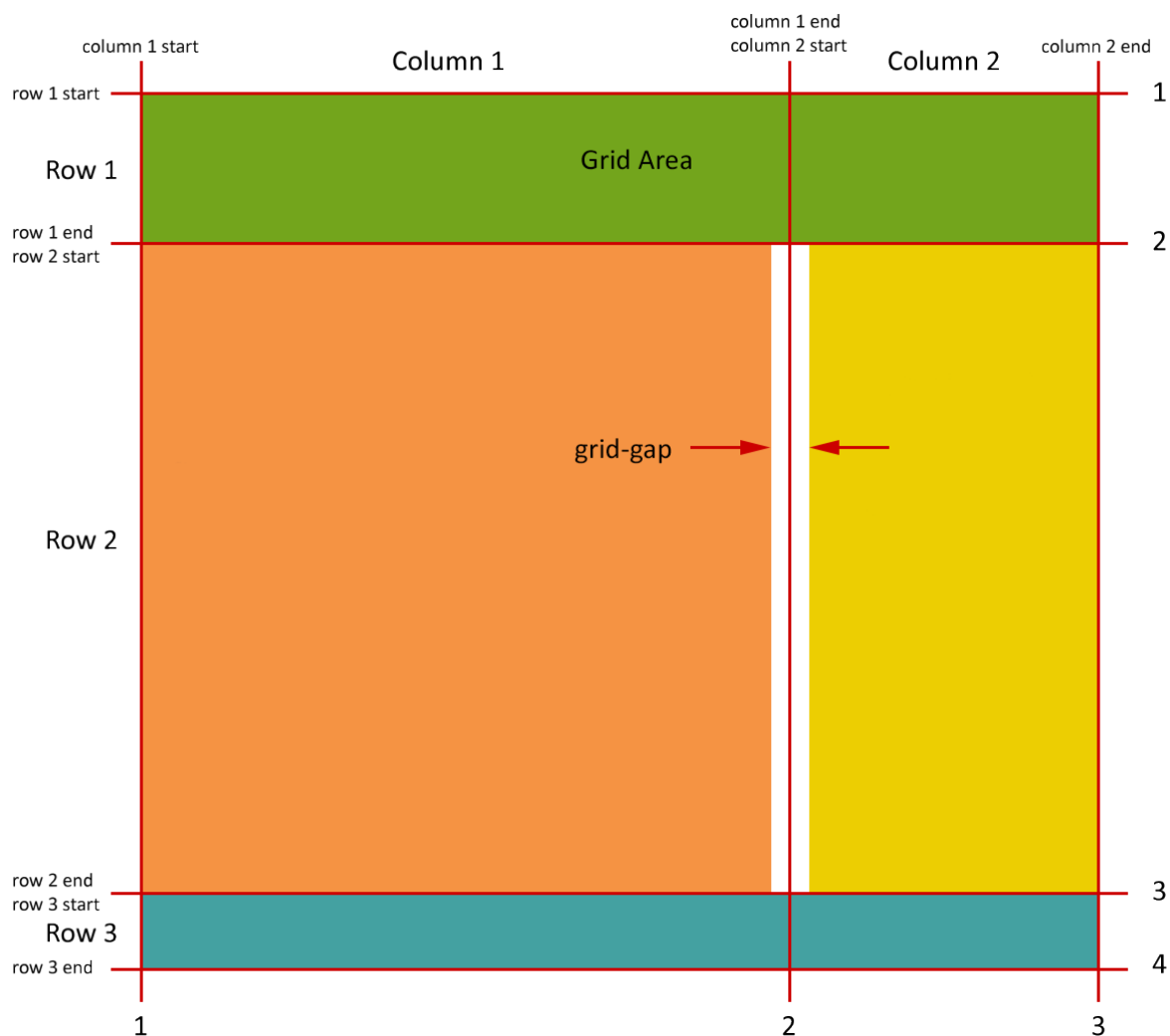
<https://www.youtube.com/layoutland>

<https://gridbyexample.com/>

<https://css-tricks.com/snippets/css/complete-guide-grid/>

<https://www.smashingmagazine.com/2017/11/css-grid-supporting-browsers-without-grid/>

CSS Grid Layout



The diagram above labels all the parts of a typical grid layout. Remember that CSS Grid was specifically designed to facilitate two-dimensional page layouts (multiple columns and rows), whereas CSS Flexbox is designed for one-dimensional (i.e. one row or one column) arrangements.

The layout above could be defined using the HTML shown on the right:

Green = <header>
Orange = <main>
Yellow = <aside>
Turquoise = <footer>

```
<div id="wrapper">

  <heading>
    <h2>This is the...</h2>
  </heading>

  <main>
    <h1>A 2 column page layout...</h1>
    <h2>The &lt;main&gt; column</h2>
    <p>Chocolate lemon drops...</p>
    <h2>A 960px wide layout...</h2>
    <p><strong>This is not...</p>
    <h2>2 columns</h2>
    <p><strong>&gt;main&lt; floated...</p>
  </main>

  <aside>
    <h2>The &lt;aside&gt; column</h2>
    <p>Biscuit brownie...</p>
  </aside>

  <footer>
    <h2>This is the...</h2>
  </footer>

</div> <!-- close wrapper -->
```

The layout could then be actioned using the CSS shown on the right:

Assuming we are using a CSS reset and that `box-sizing` is set to `border-box`, we can declare the wrapper div to be a grid container and then tell the browser where the start and end of each area is on the grid by referencing the start/end column/row.

The grid-area syntax works like this:

grid-area: row start / column start / row end / column end;

Look at the larger numbers on the diagram above and see if you can make sense of the numbers given for grid-area in the CSS on the right.

There are several syntax methods used for CSS Grid, some are easier to understand than others. Decide which one works best for you and stick with that for now.

```
* {
  box-sizing: border-box;
}
#wrapper {
  display: grid;
  grid-template-columns: 640px 300px;
  grid-gap: 0 20px;
  width:960px;
  margin:0 auto;
  border:1px black dashed;
}
header, main, aside, footer {
  margin: 1.5em;
}
header {
  height:150px;
  background-color:#73a51c;
  grid-area: 1 / 1 / 2 / 3;
}
main {
  background-color:#f59343;
  grid-area: 2 / 1 / 3 / 2;
}
aside {
  background-color:#ecce02;
  grid-area: 2 / 2 / 3 / 3;
}
footer {
  background-color:#44a1a1;
  grid-area: 3 / 1 / 4 / 3;
}
```

Class 8 Homework

Read: Chapter 14, 15 and 16 of Learning Web Design
or Chapter 11 of Head First HTML and CSS
or Chapter 17 of HTML & CSS

[Basics of CSS Grid: The Big Picture](#)

[Incredibly Easy Layouts with CSS Grid](#)

and any/all of the following:

Chapter 5 of Handcrafted CSS

Chapter 23 of Designing for the Web

Three Designed Objects project submission:

1. Complete the grid/float layout challenge we started in class if you haven't completed it.
2. Make sure your coursework homepage is online with all links working.
3. Finalise your 3 Designed Objects website (if you haven't already done so) ready for the assessment deadline on **17th November**.

Start developing the [Small Business Website project](#), bearing in mind the responses you had from your "client". Your coursework this week is to focus on the Small Business Project. Next week you will need to present your work-in-progress for this project. I recommend you continue by working up some page layout prototypes, mood boards, style tiles etc.

Course materials: [Design for Web Content](#)