

CSS – Layout

1. Graphic design on the internet

Before there were any good tools to achieve it, web designers in the early days of the internet tried to reproduce the graphic design they could achieve on printed paper. Some of the solutions used, relied on designing the page in Adobe PhotoShop and exporting it either as images to be placed in HTML tables or to Adobe's other popular tool, Flash. The result could look really nice, but the functionality became bad. Parallel to this visual emphasis ran a development of new tools to space out objects on the screen with a maintained functionality. Today, that development has given us tools that can achieve graphic designs equal and exceeding those of print. Adobe PhotoShop has become much less important as a web design tool. Many today use Figma, but writing directly in CSS is a good option for design and has the added benefit of resulting in production ready code.

[Check out Figma](#)

2. Flexbox is easier to begin with than Grid

While Flash has disappeared, many of the tools developed during the years have stayed on. So when we look for good solutions to layout problems, we are given a lot of options. In the course we will work with flexbox, which is a practical and efficient solution. In the long run, you will benefit from learning all the different tools. The most recent major tool is grid display, which gives a lot of creative possibilities. Grid display is also very good when working with responsive and adaptive web pages, that is pages that change their layout mainly on mobile phone. Grid display though has a steeper learning curve to begin with.

[Check out CSS-tricks excellent guide to Flexbox](#)

3. Spacing objects with Flexbox

Setting flex display on an element influences its immediate children.

```
display: flex;  
flex-direction: row;  
gap: 10px;  
justify-content: space-between;
```

The flex direction is by default set to row (the other option being column). The gap value influences the space between the child elements. The justify content value decides what will be done with the empty space not occupied by the child elements. For more clarification, use the link to CSS-

tricks mentioned above. That guide is really very good. Look at it in order to understand and use flexbox.

Because flexbox only works in one direction, either row or column, we have to use nested flexboxes in order to make more complex layouts.

4. Spacing objects with position

[Check out CSS-tricks entry on position](#)

A tool that can be tempting to use for site layouts is *position*. But the tool in itself can be a bit confusing and the consequences for site layouts can become fragile. Position has very good use cases in smaller and more controlled situations (imagine situations like if one wanted to place pins on a map.)

```
parent {  
  position: relative;  
}  
  
child {  
  position: absolute;  
  bottom: 30px;  
  right: -50px;  
}
```

In this case, when the parent has position relative and the child position absolute, the child element will be placed 30px up from the bottom of the parent element and 50px to the right side of it. So you can use position to break the bounding box of the parent, which makes it a powerful tool but also very problematic when it comes to site layouts (when control over parts is appreciated.)

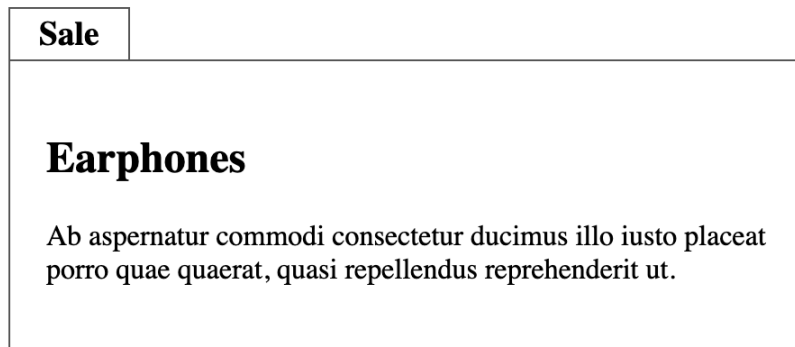
Assignment

You will work with a cloned repository from GitHub, where your role becomes to fill in the CSS code. There are three HTML pages with corresponding CSS documents. Try to style the pages so that they share the same principles as what the images below hints (judge whether a similarity is important or not for addressing a principle.)

- 1 Use *flexbox* to style a typical header navbar.

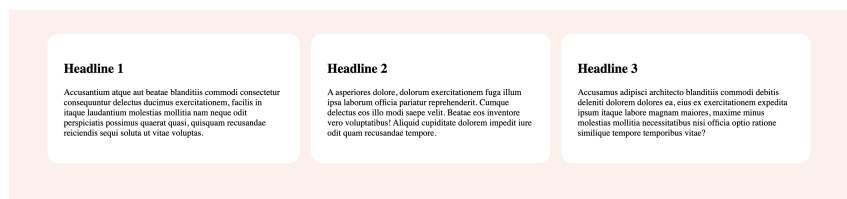
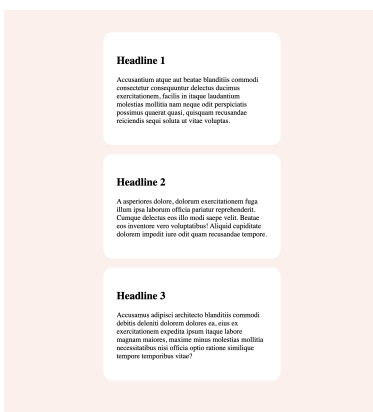


- 2 Use *position* to change the placing of the h3 element (sale).



[Check out media queries on w3schools](#)

- 3 Use *flexbox* and *media query* (included in the CSS document already) in order to get a layout that has a row direction when the screen is 900px wide or wider and column direction when the screen is narrower.



Push the repository to your own GitHub account and send the link in the chat to me at the end of the week (Friday or before next Monday morning.)