

Intermediate HTML and CSS

part 2

By Lea Ann Bradford for Girl Develop It San Diego

Three Column Content

- Why three columns?
- It's a comfortable width for readability.
- Because 3 is a pleasing design construct.

Three Column Content – HTML

- Our HTML should look something like this.

```
<section id="main-content" class="container">
  <h3>About Me</h3>
  <div id="about" class="flex-container">
    <section class="column"> <!-- First column -->
      
      <h4>Skill 1</h4>
      <p>Write a little about yourself here. </p>
    </section><!-- Repeat .column x3 -->
  </div>
</section>
```

Three Column Content – CSS

- Now that we have our 3 columns, we want them to appear next to each other. We can do this easily using Flexbox
- Flexbox makes it easier to design flexible responsive layout structure without using float or positioning

Three Column Content – CSS

- Flexbox works by creating a flex container around your content and adding your settings to that container
- In our HTML, we already added a `<div>` element with the class of “flex-container”

Three Column Content – CSS

- The parent container becomes flexible by setting the display property to *flex*
- Now that we have our flex container, we can add the flex setting in our CSS
- Wow! That was easy!

```
.flex-container {  
    display: flex;  
}
```

Three Column Content – CSS

Flex Container Properties

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

Three Column Content – CSS

Flex Direction

- defines in which direction the container wants to stack the flex items

```
.flex-container {  
    display: flex;  
    flex-direction: row;  
}
```


Three Column Content – CSS

Flex Wrap

- specifies whether the flex items should wrap or not

```
.flex-container {  
    display: flex;  
    flex-wrap: wrap;  
}
```

Three Column Content – CSS

Flex Flow

- is a shorthand property for setting both the flex-direction and flex-wrap properties

```
.flex-container {  
    display: flex;  
    flex-flow: row wrap;  
}
```

Three Column Content – CSS

Justify Content

- is used to align the flex items

```
.flex-container {  
    display: flex;  
    justify-content: space-around;  
}
```

Three Column Content – CSS

Align Items

- is used to align the flex items vertically

```
.flex-container {  
    display: flex;  
    align-items: center;  
}
```

Three Column Content – CSS

Align Content

- is used to align the flex lines

```
.flex-container {  
    display: flex;  
    align-content: space-between;  
}
```

Three Column Content – CSS

Child Element Flex Properties

- If you want an item in the child container to behave differently, you can set flex properties on it
 - order
 - flex-grow
 - flex-shrink
 - flex-basis
 - flex
 - align-self

Three Column Content – CSS

- Now let's go back and fix some of our page layout
- Now we have three columns, but they are a little too wide for our page (if you are using the side nav)
- Let's start by giving our columns a width of 30% and we can add some padding too

```
.column {  
    width: 30%;  
    padding: 15px;  
}
```

Three Column Content – CSS

- We still have a little problem with our images flowing outside of our columns
- That's because they ignore constraints like div width, unless you tell them to do so
- Let's fix that

```
.column img {  
    width: 90%;  
    max-width: 90%;  
}
```


Three Column Content – CSS

- Now let's center our images. We can do that by setting the display to block and adding an auto margin
- We can also add a border radius to make them round

```
.column img {  
    width: 90%;  
    max-width: 90%;  
    border-radius: 50%;  
    margin: 0 auto 20px auto;  
    display: block;  
}
```

Let's Develop It

- Let's work on three column section
- Adjust the font size of the main header of the content area
- Adjust the font size of the headers in the columns - try changing their colors too
- Add a border to the images to make them stand out a bit more

Experience Section – More Flexbox

- Let's create a section in our page that shows our job history, like you would have on your resume
- We can dive a little deeper into flexbox for the layout

Experience Section – HTML

- Our HTML should look something like this.
- Notice the class names? Yep, we're gonna use some flex!

```
<section id="skills" class="container">
  <h3>My Experience</h3>
  <section class="job"> <!-- First job in list -->
    <p class="flex-basis-25">
      My Company Name<br>My Title
    </p>
    <ul class="flex-basis-50">
      <li>List of your responsibilities</li>
    </ul>
    <p class="flex-basis-25">Start date – End date</p>
  </section> <!-- Repeat .job for each position -->
</section>
```

Experience Section – CSS

- Let's add the flex property to our .jobs class
- Let's justify our content using the space around property
- Finally, let's add some padding and a bottom border

```
.job {  
    display: flex;  
    justify-content: space-around;  
    padding: 24px 0;  
    border-bottom: 1px solid #e6ac00;  
}
```

Experience Section – CSS

- This still doesn't look quite right.
- Let's adjust the width of our individual flex items

```
.flex-basis-25 {  
    flex-basis: 25%;  
}  
  
.flex-basis-50 {  
    flex-basis: 50%;  
    padding-right: 16px;  
}
```

Experience Section – CSS

- I like to add my bullet items back to my `` list items for this section
- Let's try adding the circle bullet

```
.job ul {  
    list-style-type: circle;  
}
```


Some References – Flexbox

More on Flexbox and to each of the properties in action:

○ <https://codepen.io/enxaneta/pen/adLPwv>

To play a game and learn flexbox:

○ <https://flexboxfroggy.com/>

Icons and Fonts

- The world of HTML has progressed beyond Times New Roman and Arial
- How do we use modern and stylish fonts?
- And how do we add cool icons?

Google Fonts

- Meet Google Web Fonts!
- Google has hundreds of free, open-source fonts that have been optimized for the web, and ready for us to use!
- The service runs on Google's servers which are fast, reliable and tested. Google provides this service free of charge.

Google Fonts

- www.google.com/fonts

- In our example, we've used Lato for the body and Bree Serif for the headlines
- You can use any font you'd like!

Google Fonts

- Three easy steps:

1. Search for the fonts you want and add them to your collection
2. Narrow down your results – only choose what you will need (less is better)
3. Grab the code that Google prepares for you and add it to your site!

```
@import url(http://fonts.googleapis.com/  
css?family=Lato:300,400,700,300italic,400italic|Bree+Serif);
```

Google Fonts

- Now let's add your fonts to the CSS

```
@import
url(http://fonts.googleapis.com/css?family=Lato|
Bree+Serif);

body {
    font-family: 'Lato', sans-serif;
}

h1, h2, h3, h4, h5, h6 {
    font-family: 'Bree Serif', serif;
}
```

Font Awesome Icons

- <https://fontawesome.com/>
- <https://fontawesome.com/cheatsheet>
- Font Awesome Free is open source and completely free for commercial use.
- Font Awesome has over 900 icons for you to choose from
- Font Awesome icons have accessibility built in

Font Awesome Icons

○ Three easy steps:

1. Add the link to the Font Awesome CDN to the <head> section of your HTML
2. Search for the icon you want to use
3. Add the icon code to your HTML & style it with CSS

```
<script defer src="https://use.fontawesome.com/releases/v5.0.10/js/all.js"
integrity="sha384-slN8GvtUJGnv6ca26v8EzVaR9DC58QEwsIk9q1QXdCU8Yu8ck/tL/5szYlBbqmS+"
crossorigin="anonymous"></script>
```

Font Awesome Icons

- Let's add a section with icons that will link to your social media
- Your HTML should look something like this

```
<section id="social-links">  
  <a href="http://www.mylinkedin.com">  
    <i class="fab fa-linkedin-in fa-3x"></i>  
  </a>  
</section>
```


Font Awesome Icons

- Now let's add some CSS to your social section
- Let's give your social section a background color, some margin and padding
- Let's also give some margin and color to your icons (target the `<a>` tag)

```
#social-links {  
    background: #8a80c3;  
    margin-top: 20px;  
    padding: 25px 0;  
    text-align: center;  
}  
#social-links a {  
    margin: 20px;  
    color: #fafafa;  
}
```

Let's Develop It

- Let's work fonts and your social section
- Pick some fonts for your site using Google Fonts.
- Adjust the font-size and line-height to refine your fonts.
- Add links and icons for your social media sites
- Style the social section and icons to your liking

Questions?