Style Conference Part VI

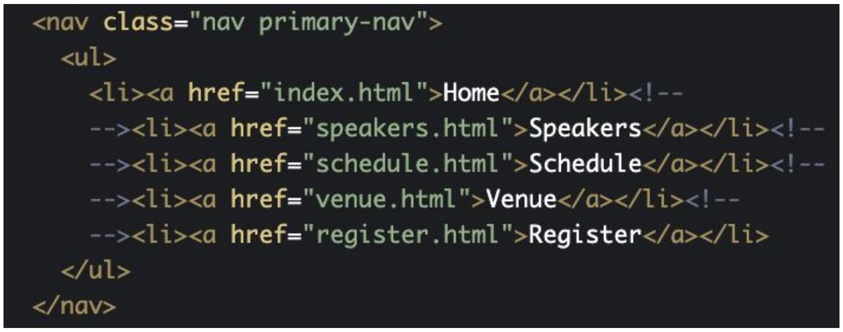
AIM Code School

Now that we know how to build lists within HTML and CSS, let’s loop back to our Styles Conference website and see where we might be able to use lists.

Using an unordered list (via the **<ul>** element) and list items (via the **<li>** element) will give structure to our navigation menus. These new elements, however, will display our navigation menus vertically.

We’re going to want to change the display value of our **<li>** elements to inline-block to get all of them to align in a horizontal row. When we do that, though, we’ll also need to account for the blank space left between each **<li>** element.

Along these same lines, the markup for the navigation menu within the **<footer>** element will look similar however the class name on the nav tag is “nav”.

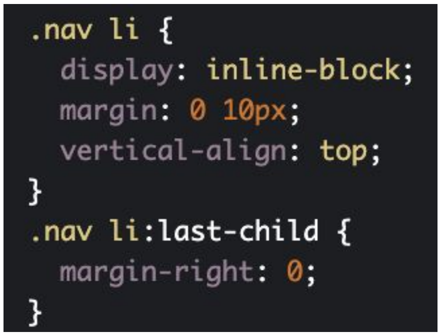


```
<nav class="nav primary-nav">
  <ul>
    <li><a href="index.html">Home</a></li><!--
    --><li><a href="speakers.html">Speakers</a></li><!--
    --><li><a href="schedule.html">Schedule</a></li><!--
    --><li><a href="venue.html">Venue</a></li><!--
    --><li><a href="register.html">Register</a></li>
  </ul>
</nav>
```

DON'T FORGET TO MAKE THESE CHANGES ON ALL OF YOUR HTML PAGES FOR THIS WEBSITE!

Let's move on to the CSS. We'll start by setting all of the **<li>** elements within any element with the class attribute value of **nav** to be displayed **inline-block**, to include some horizontal margins, and to be vertically aligned to the **top** of the element. Under our Navigation rule set add:

```
.nav li {
  display: inline-block;
  margin: 0 10px;
  vertical-align: top;
}
.nav li:last-child {
  margin-right: 0;
}
```



```
.nav li {
  display: inline-block;
  margin: 0 10px;
  vertical-align: top;
}
.nav li:last-child {
  margin-right: 0;
}
```

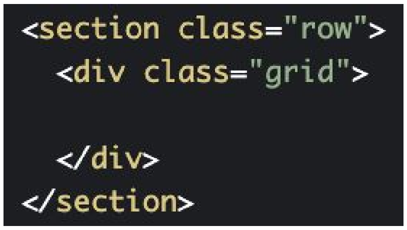
*You may be wondering why our unordered list didn't include any list item markers or default styles. These styles were removed by the reset at the top of our style sheet.

We're also going to be using lists on the speaker page. So go ahead and let's add some speakers to our conference.

Within our **speakers.html** file just below our lead section, let's create a new section where we will present all of our speakers. Reusing some existing styles, well use a **<section>** element with a class attribute value of row to wrap all of our speakers and apply a white background and padding behind them. Inside the **<section>** element, we'll add a **<div>** element with a class attribute value of grid to center our speakers on the page and allow us to use multiple columns in doing so.

So far our HTML below the lead section looks like this:

```
<section class="row">

  <div class="grid">



  </div>

</section>
```

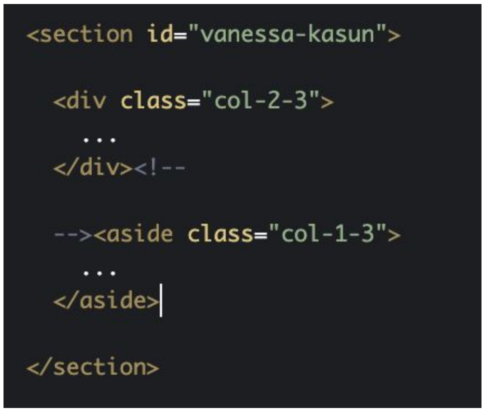


```
<section class="row">
  <div class="grid">

  </div>
</section>
```

Inside the grid section, every speaker will be marked up with his or her own **<section>** element, which will include two columns. The first column will span two-thirds of the **<section>** element and will be marked up using a **<div>** element. The second column will span the remaining one-third of the **<section>** element and will be marked up using an **<aside>** element, as its content is secondary to the speaker and his or her specific talk.

Using our existing **col-2-3** and **col-1-3** classes, the outline for a speaker section will look like this:

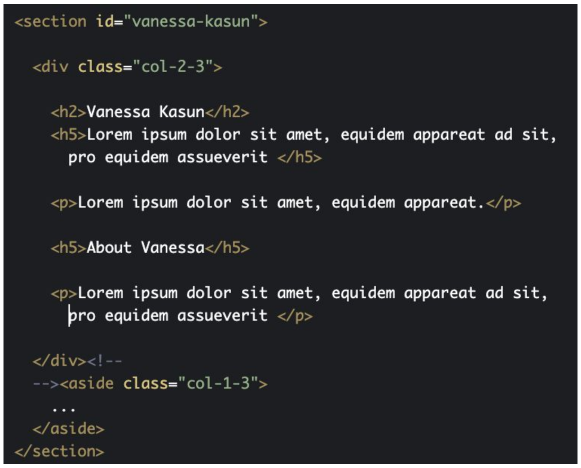
```
<section id="vanessa-kasun">

  <div class="col-2-3">

    …

  </div><!--

  --><aside class="col-1-3">

    …

  </aside>

</section>
```



```
<section id="vanessa-kasun">

  <div class="col-2-3">
    ...
  </div><!--

  --><aside class="col-1-3">
    ...
  </aside>

</section>
```

Inside the two-thirds column, marked up with the <div> element, we'll use a few headings and paragraphs to show the speaker's name, the title and abstract of the talk, and a short biography.

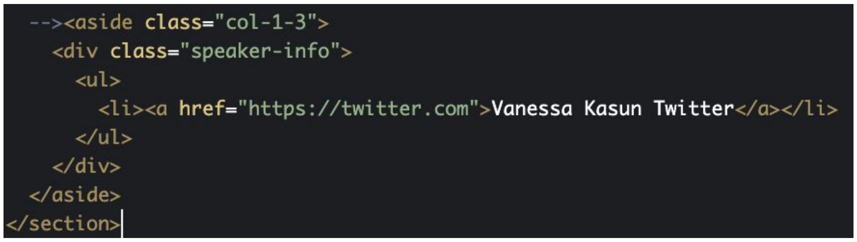
```
<section id="vanessa-kasun">


 <div class="col-2-3">

   <h2>Vanessa Kasun</h2>

   <h5>Lorem ipsum dolor sit amet, equidem appareat ad sit,

     pro equidem assueverit </h5>

   <p>Lorem ipsum dolor sit amet, equidem appareat.</p>

   <h5>About Vanessa</h5>

   <p>Lorem ipsum dolor sit amet, equidem appareat ad sit,

     pro equidem assueverit </p>

 </div><!--

 --><aside class="col-1-3">

   ...

 </aside>

</section>
```



```
<section id="vanessa-kasun">

  <div class="col-2-3">

    <h2>Vanessa Kasun</h2>
    <h5>Lorem ipsum dolor sit amet, equidem appareat ad sit,
      pro equidem assueverit </h5>

    <p>Lorem ipsum dolor sit amet, equidem appareat.</p>

    <h5>About Vanessa</h5>

    <p>Lorem ipsum dolor sit amet, equidem appareat ad sit,
      pro equidem assueverit </p>

  </div><!--
  --><aside class="col-1-3">
    ...
  </aside>
</section>
```

Within the one-third column, marked up with an **<aside>** element, we're going to add a **<div>** element with a class attribute value of **speaker-info**. We'll use a **<div>** element because we'll be adding styles to this element soon.

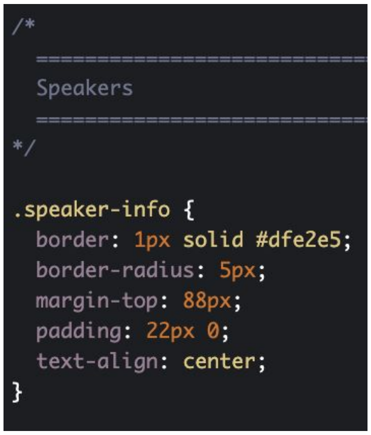
Before getting into any styles, though, let's add an unordered list within the **<div>** element that includes as list items some relevant links for the speaker.



```
  --><aside class="col-1-3">
    <div class="speaker-info">
      <ul>
        <li><a href="https://twitter.com">Vanessa Kasun Twitter</a></li>
      </ul>
    </div>
  </aside>
</section>
```

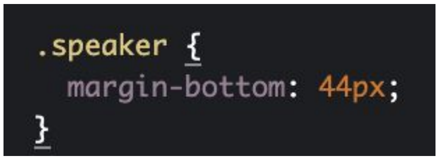
Let's switch gears and go to our **main.css** file. Make a comment rule set for the **speakers** section and add the following selector with the corresponding property value pairs:

```
.speaker-info {
  border: 1px solid #dfe2e5;
  border-radius: 5px;
  margin-top: 88px;
  padding: 22px 0;
  text-align: center;
}
```



```
/*
  ============================
  Speakers
  ============================
*/

.speaker-info {
  border: 1px solid #dfe2e5;
  border-radius: 5px;
  margin-top: 88px;
  padding: 22px 0;
  text-align: center;
}
```

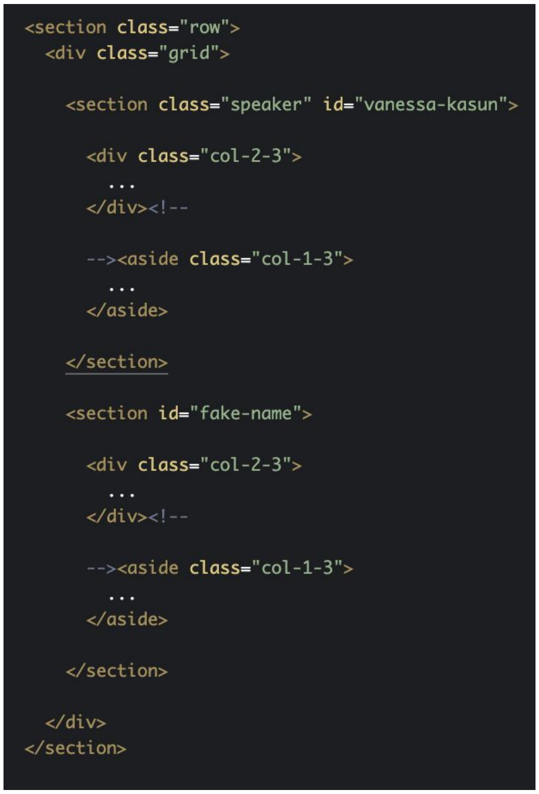
We're placing a **<div>** element inside the **<aside>** element with the class attribute value of **col-1-3** because we'll want the padding inherited from the **col-1-3** class to be outside of the border on the **<div>** element. Before long we'll be including an image within the **<div>** element, alongside the unordered list; therefore we created a **<div>** element as opposed to applying these styles directly to the **<ul>** element.



```
.speaker {
  margin-bottom: 44px;
}
```

We can then apply this class to the **<section>** element for each speaker, provided it isn’t the last speaker. We’ll omit this class on the last speaker, as we don’t want to create any unnecessary margins before our **<footer>** element. With more than one speaker, our layout will look like this:

```
<section class="row">
 <div class="grid">
  <section class="speaker" id="vanessa-kasun">
   <div class="col-2-3">
...
   </div><!--
   --><aside class="col-1-3">
...
   </aside>
  </section>
  <section id="fake-name">
   <div class="col-2-3">
...
   </div><!--
    --><aside class="col-1-3">
...
   </aside>
  </section>
 </div>
</section>
```



```
<section class="row">
  <div class="grid">

    <section class="speaker" id="vanessa-kasun">

      <div class="col-2-3">
        ...
      </div><!--

      --><aside class="col-1-3">
        ...
      </aside>

    </section>

    <section id="fake-name">

      <div class="col-2-3">
        ...
      </div><!--

      --><aside class="col-1-3">
        ...
      </aside>

    </section>

  </div>
</section>
```

Notice how the first speaker **<section>** element, for Vanessa Kasun, includes the class attribute value of **speaker**, which vertically separates it from the speaker **<section>** element for **fake-name**. The last speaker **<section>** element, again for fake-name, doesn’t include a class attribute value of speaker in order to keep it a proper distance from the <footer> element.

Our navigation menus are now complete, and the speakers page is taking shape!

Download the Source Code Zip Folder to double check for any error in your code if yours is not rendering correctly.

SOURCE CODE