Module 8 Response Paper

One point where flexbox and CSS grid diverge is the template that you create in grid using “fr”, a very flexible length unit that stands for fraction. If you define 3 fr, all three are equal width; if you define 6 then all 6 are equal width. This is true for all fr’s until otherwise defined. These lines automatically now have numbers. Each fr is a number, with the first colmin being 1, the second being 2, etc. Right now, trying to load this would be very unfruitful because there is no content inside of these and we have not even specified rows yet. The css is not hard for these things, creating the grid is simply setting the parent:

```
display: grid;
grid-template-columns: 1fr 1fr 1fr;
```

Then to set our children to know where they go:

```
grid-column: 1 / 3;
grid-row: 1;
```

This is where things can get tricky. Grid-column, to the untrained eye, can be a problem because you may think “well this starts on column 1 and ends on column 3. But this is not the case, this actually reads that it starts on column 1 but ends at column 3 and therefore, column three is still open. Your next style may look like:

```
grid-column: 3;
grid-row: 1;
```

This is taking up the last room in that column so now the row is filled. Named areas are super powerful as well, you can name an item and then lay it out using grid-template-areas.