college CDI

Layout fundamentals for websites (NMW)

Class 05

Creating a CSS layout grid

12 columns layout grid

Once a sketch has been selected and that the Alpha version of a website has been produced, it is time to start integrating the pages contents. As it was priorly established, the use of a grid will assure cohesion an will greatly facilitate the developer's work.

Step 1 : The grid container

In a HTML5 page structure, create a container which will be the grid to be used. There certainly are many different ways to create a layout grid using CSS, but the easiest way probably is to use the grid display which will divide the pages width in twelve equal parts with a 1rem gutter.

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
</head>
<body>
<div class="grid">

</div>
</body>
</html>
```

CSS

```
.grid {
        display: grid;
        grid-template-columns: repeat(12, 1fr);
        grid-column-gap: 1rem;
        grid-row-gap: 1rem;
}
```

NOTA :
For a 960px (or 1200px, etc.), simply add a *width: 100%* to the grid and a *margin: auto* so it is centered horizontal. For such grids, the gutter is usually set to 20px.

For a 16 columns grid, simply use *16* instead of *12* in the repeat function.

Step 2 : Spanning items horizontally (over columns/units)

Now, all there is to be done is to add items into the grid container and span them over the wanted number of units (columns).

HTML

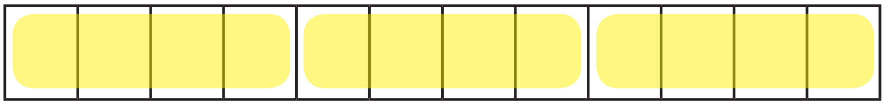
```
<div class="grid">
        <header> </header>
</div>
```

CSS

```
header {
        grid-column: span 12;
}
```

NOTA :
With some techniques, rows and columns need to be created. Here, when the sum of the units reach twelve (or sixteen, depending on the type of grid used), a new row starts automatically.

Example :
For 3 items to use 4 units each, the items would need grid-column: span 4, and the 3 items would remain on one row.



HTML

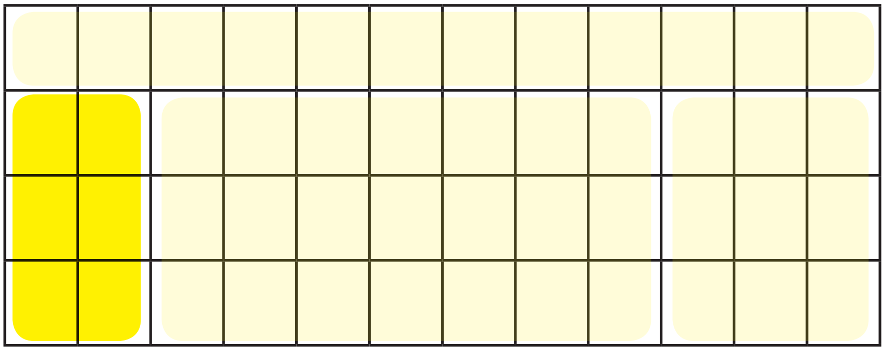
```
<div class="grid">
     <div class="item"> </div>
     <div class="item"> </div>
     <div class="item"> </div>
</div>
```

CSS

```
.item {
          grid-column: span 4;
}
```

Spanning items vertically (over rows)

Whenever it is needed for an items to span vertically over rows, ***grid-row*** property must be used. For instance, for an item to be spent over 3 rows, ***grid-row: span 3*** would be used.



HTML

```
<div class="grid">
     <div class="item"> </div>
     <div class="item2"> </div>
     <div class="item"> </div>
     <div class="item"> </div>
</div>
```

CSS

```
.item2 {
     grid-column: span 2;
     grid-row: span 3;
}
```

General advises

Responsive grids

In order for a grid to be responsive, the use of a media-query will allow, for instance, for all the items to be displayed one over the other in a single column using a flex display.

```
@media only screen and (max-width: 768px) {
        .grid {
                display: flex;
                flex-direction: column;
        }
}
```

Margin

Be careful to have sufficient gutter space so there is clear separations between the different zones created. Also, some container will need to have different margin adjustments made so texts align correctly (especially for titles).

Headings

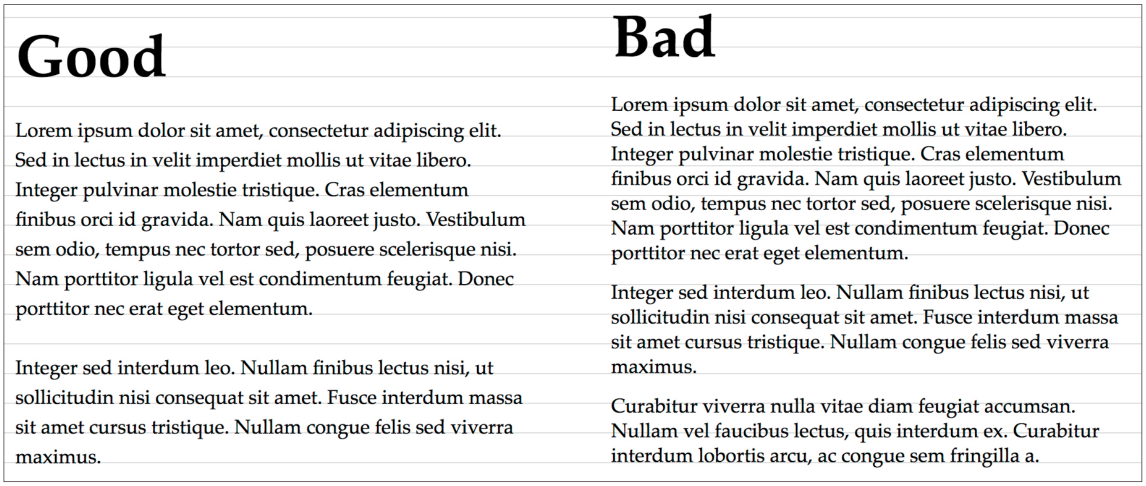
Headings such as <h1>, depending on the browsers, may have padding and margin. Make sure to reset those to zero to avoid unnecessary corrections.

Nested grids

Although an entire site is built within a main grid, it often happen for a section (for instance the main section) to be divided into a grid, especially for website showing high volume of content.

Baseline grid

Whenever texts are shown side by side in two different container, make sure to use a baseline alignment in order for all text lines in the page to be correctly aligned.



Good

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed in lectus in velit imperdiet mollis ut vitae libero. Integer pulvinar molestie tristique. Cras elementum finibus orci id gravida. Nam quis laoreet justo. Vestibulum sem odio, tempus nec tortor sed, posuere scelerisque nisi. Nam porttitor ligula vel est condimentum feugiat. Donec porttitor nec erat eget elementum.

Integer sed interdum leo. Nullam finibus lectus nisi, ut sollicitudin nisi consequat sit amet. Fusce interdum massa sit amet cursus tristique. Nullam congue felis sed viverra maximus.

Bad

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed in lectus in velit imperdiet mollis ut vitae libero. Integer pulvinar molestie tristique. Cras elementum finibus orci id gravida. Nam quis laoreet justo. Vestibulum sem odio, tempus nec tortor sed, posuere scelerisque nisi. Nam porttitor ligula vel est condimentum feugiat. Donec porttitor nec erat eget elementum.

Integer sed interdum leo. Nullam finibus lectus nisi, ut sollicitudin nisi consequat sit amet. Fusce interdum massa sit amet cursus tristique. Nullam congue felis sed viverra maximus.

Curabitur viverra nulla vitae diam feugiat accumsan. Nullam vel faucibus lectus, quis interdum ex. Curabitur interdum lobortis arcu, ac congue sem fringilla a.