

## CSS Properties discussed in class

Below is a brief synopsis of the properties discussed in class. For more information such as additional property values visit [www.w3schools.com/cssref/default.asp](http://www.w3schools.com/cssref/default.asp). For a complete list of current **browser support** visit [www.w3schools.com/cssref/css3\\_browsersupport.asp](http://www.w3schools.com/cssref/css3_browsersupport.asp)

Description	Example
<b>display: grid;</b> Defining the display property as a grid tells the browser to treat the div container as a grid.	<pre>.grid-container {   display: grid; }</pre>
<b>grid-gap property.</b> The grid-gap property defines the of space between grid items both horizontally and vertically. <ul style="list-style-type: none"> <li>• <b>grid-column-gap: value;</b> – column spacing</li> <li>• <b>grid-row-gap: value;</b> – row spacing</li> <li>• <b>grid-gap: columnvalue rowvalue;</b> – shorthand for both with different values</li> </ul> <b>grid-gap: value;</b> – shorthand for both if they have the same value	<pre>.grid-container {   grid-gap: 10px; }</pre>
<b>grid-template-areas: property.</b> Defining the grid area using names can make it easier to easily see the pattern of your grid by laying out the pattern of the grid using named values. <b>grid-template-areas:</b> <pre>'name1 name1 name1' 'name2 name3 name3' 'name2 name3 name3';</pre>	<pre>.grid-container {   grid-template-areas:     'mainnav mainnav mainnav '     'maingallery maingallery maingallery'     'footer footer footer'; }</pre>
<b>grid-area property.</b> The grid-area defines what you want your grid item to be named. Using these names, you can create a pattern for your div using the grid-template-areas property. <b>grid-area: name;</b>	<pre>.grid-mainnav {   grid-area: mainnav; }</pre>
<b>display: flex;</b> By declaring the display to be the flex value a flexbox is created. The flexbox styles allow you to easily create various responsive layouts without having to use the float property.	<pre>.gallery-row {   display: flex; }</pre>
<b>Flexbox-direction property.</b> The flexbox-direction property sets the main axis or how the items will be laid out either horizontally or vertically. <ul style="list-style-type: none"> <li>• <b>flex-direction: row;</b> sets the main axis to be a single row of columns</li> <li>• <b>flex-direction: row-reverse;</b> sets the main axis to be a single row of columns that runs in the reverse direction</li> <li>• <b>flex-direction: column;</b> sets the main axis to be a single column of rows</li> <li>• <b>flex-direction: column-reverse;</b> sets the main axis to be a single column of rows that runs in the reverse direction</li> </ul>	<pre>.gallery-row {   display: flex;   flex-direction: row; }</pre>

**flex-wrap property.** Declares that the flex items can wrap to the next line.

```
.gallery-row {  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
}
```

**Flex property.** Flex is shorthand for flex-grow, flex-shrink and flex basis. *flex: 100%*; sets grow, shrink, basis properties to 100%

```
.gallery-column {  
  flex: 100%;  
}
```

**Max-width.** Declares the maximum width of a element.

```
.gallery-column {  
  flex: 100%;  
  max-width: 100%;  
}
```