

Flexbox and Grid are CSS techniques that have enormously simplified many layout tasks in HTML; and, in fact, have made it possible to do things that were previously not possible.

FLEXBOX / "FLEX"

ONE-DIMENSIONAL LAYOUT CONTROL Flex is good for arranging elements either horizontally **or** vertically. It's a good alternative to floats when it comes to layout (floats should be left for creating small text wraps in most cases). With flexbox, the parent element becomes the flex container; child elements become flex items.

FLEXBOX: CSS SYNTAX The parent element has to have two properties designated in the CSS to work:

- 1) The parent selector must be set to `display: flex;`
- 2) The parent must also have a flex-direction value declared. There are four basic options for this (to start with): row, row-reverse, column, and column-reverse, as in:



```
<div id="container">
  
  <h2> Hello world </h2>
  <p> Lorem ipsum paragraph here... </p>
</div>

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

CSS GRID / "GRID"

TWO-DIMENSIONAL LAYOUT CONTROL Grid enables the horizontal **and** vertical organization of elements across a true grid (use of columns **and** rows). Enormous layout potential with much less code than other methods. The parent element is the grid container; child elements (direct descendents) become grid items.

GRID: CSS SYNTAX The parent element has to have two properties designated in the CSS for grid to work:

- 1) The parent selector must be set to `display: grid;`
- 2) The parent must also have a values for the grid divisions to be declared. This can be achieved a wide number of ways. The simplest way to start is to assign values for the columns, rows and grid gaps, as in:



```
<div id="container">
  
  
  
  
  
  
</div>

.container {
  display: grid;
  grid-template-columns: 1fr 2fr 1fr;
  grid-template-rows: auto;
  grid-gap: 10px; }
```

IMAGES

Images can either be part of the HTML content, (ex: image of a product for sale); or, images can be presentational from within CSS (ex: a mood-setting background image).

IMAGE FILE FORMATS AND DETAILS

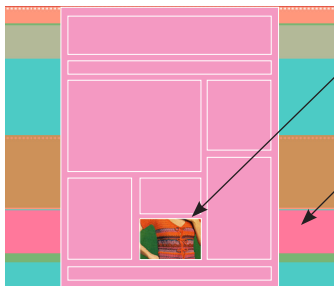
- **PNG** (Portable Graphics Interchange Format files, .png extension) — a bitmap file format designed for use on the internet that uses lossless compression; intended to replace the GIF format
- **GIF** (Graphics Interchange Format files, .gif extension) — for cartoons, logos, graphics with transparent areas, and animations. GIFs contain a maximum of 256 colors
- **JPEG/JPG** (Joint Photographic Experts Group files, .jpg extension) — usually photographs or high-color images; best for digital or scanned photographs, images using textures, images with gradient color transitions, and any images that require more than 256 colors
- **SVG** (scalable vector graphic, .svg extension) — vector matter from Illustrator; scales well; displays well on high-res screens

Color Mode = RGB or hex values always.

Image resolution = in pixels (72 dpi; but higher lately: 150 dpi); size optimization is critical! The smaller the overall file size, the better.

OPTIMIZING IMAGES FOR THE WEB All image files should be optimized for web display; that is, they should be saved as small as necessary to facilitate speedy display, but as high as possible to ensure the desired quality. In Photoshop, use File > Export... > “Save for Web (Legacy)...”. This allows you to compare lossy results prior to saving. Save files with a consistent, clear naming system when they are the final, web-ready file.

INSERTING IMAGES IN DREAMWEAVER



HTML content images To insert *searchable content images* into the HTML source code, use Insert > Image > (then browse to desired image). The HTML tag looks like this:
``

CSS presentational images To insert *non-searchable images and graphics* that express brand “mood”) — use CSS. Background images are a common example. A rule can look something like this:

```
body {  
    background-image: url(images/stripes_pattern.png);  
    background-repeat: repeat; }  
}
```

IMAGE SIZING

First, Photoshop image files should be sized with an approximate use size in mind. Second, since image sizing is a presentational layout decision, images are best sized for layout placement via CSS. There is a wide variety of techniques for placing images where you want them at the right size, such as:

```
img {  
    width: 100%;  
    height: auto;  
}  
#section img {  
    width: 100%;  
    height: auto;  
}  
#section img {  
    width: 50%;  
    height: auto;  
    float: left;  
}
```