Back-end

* files that are interpreted by server
* Server doesn’t care what the string is as long as it’s a string
* Static files 🡪 strings
* Usually .rb files

Front-end

* files that are interpreted by client
* usually .html, .css, .js
  + designed to do less
  + The separation of concerns

Dynamic\_html

Metaprogramming

* code that writes other code

HTML

CSS

* float
  + inside the content = responsive programming
  + must be before the text which it is beside
* space
  + HTM does not recognize spacing in code, outside of your tags.
  + Magins can but up against each other
    - You can use negative numbers to adjust : -30px
* Fluid vs. responsive
* The main difference is that Fluid Layouts (also called Liquid Layouts) are based on proportionally laying out your website so elements take up the same percent of space on different screen sizes, while Responsive Design uses [CSS Media Queries](http://kyleschaeffer.com/best-practices/responsive-layouts-using-css-media-queries/) to present different layouts based on screen sizes/type of screen. For some examples of both kinds of design, see [Inspiration: Fluid & Responsive Design](http://webdesignerwall.com/trends/inspiration-fluid-responsive-design).
* Fluid's intent is to keep the same spatial weighting to all elements, and works okay on different sizes of screens of the same sort. They tend to look okay on a 32", high resolution monitor and a 12" lower resolution laptop. They're pretty easy to implement.
* Responsive design's intent is to serve different devices layouts tailored specifically for the type of screen. Your site's layout will generally be cut down to a single column on a smartphone for example.

When to use JPEG vs. GIF

JPEG is **not** going to displace GIF entirely. For some types of images, GIF is superior in image quality, file size, or both. One of the first things to learn about JPEG is which kinds of images to apply it to.

Generally speaking, JPEG is superior to GIF for storing full-color or grey-scale images of "realistic" scenes; that means scanned photographs and similar material. Any continuous variation in color, such as occurs in highlighted or shaded areas, will be represented more faithfully and in less space by JPEG than by GIF.

GIF does significantly better on images with only a few distinct colors, such as line drawings and simple cartoons. Not only is GIF lossless for such images, but it often compresses them more than JPEG can. For example, large areas of pixels that are all **exactly** the same color are compressed very efficiently indeed by GIF. JPEG can't squeeze such data as much as GIF does without introducing visible defects. (One implication of this is that large single-color borders are quite cheap in GIF files, while they are best avoided in JPEG files.)

Computer-drawn images (ray-traced scenes, for instance) usually fall between photographs and cartoons in terms of complexity. The more complex and subtly rendered the image, the more likely that JPEG will do well on it. The same goes for semi-realistic artwork (fantasy drawings and such).

JPEG has a hard time with very sharp edges: a row of pure-black pixels adjacent to a row of pure-white pixels, for example. Sharp edges tend to come out blurred unless you use a very high quality setting. Edges this sharp are rare in scanned photographs, but are fairly common in GIF files: borders, overlaid text, etc. The blurriness is particularly objectionable with text that's only a few pixels high. If you have a GIF with a lot of small-size overlaid text, don't JPEG it.