Graphics and Fonts

Lecture 16
Colors in CSS

- Colors used for fonts, borders, background
- RGB as decimal (0-255), percentage, or hex
  - `rgb (255,0,0) /*pure red*/`
  - `rgb (100%,0%,0%)`
  - `#ff0000`
- HSL (Hue, Saturation, Light)
  - Hue (0-360) is angle on color wheel: 0 is red, 120 green, 240 blue
  - Saturation & light are both %'s
    - `hsl (0,100%,50%) /*full bright red*/`
- Alpha channel (transparency): 1 is opaque!
  - `rgba (255,0,0,0.5)`
  - `hsla (0,100%,50%,1)`
RGB Cube

http://www.flickr.com/photos/ethanhein/3103830956/
HSL Color Wheel (100% Sat.)

HSL Grid for Red (ie 0,x,y)

- (0,75%,88%)
- (0,100%,50%)
- (0,0%,25%)
Color Depth

- "Depth" = # of bits in representation
  - 8 bits → 256 different colors
  - 24 bits → 16,777,216 different colors (eg 8 bits each for r,g,b)

- Alpha may be (incorrectly) included
  - rgba is a point in 4-dimensional space

- Problem: image color depth > display color depth
  - Quantization: each pixel gets closest available color (leads to banding)
  - Dithering: add noise, which looks better!
Quantization vs Dithering

- Original
- Quantized
- Dithered
Quantization vs Dithering

Original Image  GIF without dithering  GIF with dithering

HTML `<img>` Tag Attributes

- **src:** location of image file
- **width/height:**
  - Area in `window` to reserve for image
  - Image is `scaled` to those dimensions
  - These attributes affect browser flow, regardless of when/if image is displayed
- **alt:** text to show if graphic cannot be displayed or seen (i.e., alternative)
- **title:** text to `augment` displayed graphic (e.g., tooltip)
Image Representation

- Raster vs vector graphics
  - Raster: stored pixel-by-pixel
  - Vector: mathematical description

- Compression of raster images
  - Lossy: better compression, lower quality image
  - Lossless: largest file size, best quality
Major Formats

- **JPEG**
  - Raster, lossy compression
  - 24 bit, no transparency
  - Good for photos, gradual gradients

- **PNG**
  - Raster, lossless (often high) compression
  - Variable depth, has transparency
  - Good for icons and crisp lines

- **GIF**
  - Raster graphics, lossy compression (oldest)
  - 8 bit (ie very small files)
  - Frame-based animation (please avoid!)

- **SVG**
  - Vector graphics (newest)
Scaling Images

- Vector graphics scale perfectly

- Raster images should be *pre-scaled*
  - Width (height) attributes of image tag should match actual width (height) of image
  - Why?
Alternative: CSS

```css
.deleteButton {
  background: -webkit-linear-gradient(top, #be6868 0%, #941b17 50%, #880d07 50%, #be483c 100%);
  border: 3px solid #000;
  color: #fff;
  cursor: pointer;
  font-size: 15pt;
  padding: 10px 34px;
  text-shadow: 0 -1px 0 #000;
  border-radius: 13px;
  box-shadow: 0 1px 0 #454545;
}
```
Recall: Blocks, Inline, and Flow
Floating: Remove From Flow

width
Floating: Overlays Block
Problem: Blocks Below

- Floating element may be taller than containing element
- Undesirable, e.g., for footer that should be below everything including floats
Solution: clear

- Styling for block element after float
  ```
  #footer { clear: left; }
  ```

- Requires that side to be clear of floats
CSS Units for Size

- "Absolute" units (but browsers cheat)
  - in, cm, mm
  - pt (point) = 1/72 inch, pc (pica) = 12 pts

- Absolute (for a given resolution)
  - px (pixels)

- Relative to current font
  - em = width of 'm' in current font
  - ex = height of 'x' in current font

- Relative to parent (or ancestor) size
  - %

- Standard advice for fonts:
  - Prefer % (or em) for user-controlled scaling
Aside: The Problem with Pixels

- Historically, pixel size determined by *hardware* (ie screen resolution)
  - ppi: "pixels per inch"
- Problems using px unit:
  - Different resolutions = different sizes
  - Different devices = different view distances
- Solution: W3C's "reference pixel" (*optics*)
Fonts: Concepts

- Fonts are a key part of visual design
  - Serious, technical, whimsical, friendly...
- Font families (should be "typefaces")
  - Arial, Helvetica, Times, Courier, Palatino, Garamond, Verdana, Tahoma, Lucida,...
- Serif vs sans-serif
- Variable vs monospace letter spacing
- Line spacing and x-height
  - Larger x-height = easier to read
  - Larger x-height requires more line spacing
Font Families

- *De gustibus non est disputandum*

- Still, some common opinions:
  - Less is more: Use fewer fonts/sizes
    - Cohesive appearance
  - Helvetica/Arial: clean but ubiquitous
    - They are identical / completely different
  - Times is hard to read (on a monitor)
    - Better for print
  - Comic Sans is for 12-year-olds and owners of NBA basketball teams
Identical & Completely Different

Arial vs Helvetica

http://typographytoday.posterous.com
Fallback Fonts

- Not sure what fonts host OS will have
- CSS font-family: List alternatives in decreasing order of preference
  ```
  font-family: Helvetica, Arial, "Liberation Sans", sans-serif;
  ```
- Always end with one of 5 generic fonts:
  - sans-serif (Arial?)
  - serif (Times New Roman?)
  - monospace (Courier New?)
  - cursive (Comic Sans?)
  - fantasy (Impact?)
- OS (and browser) determine which font family each generic actually maps to
CSS3: Web Fonts @font-face

- Looks like a selector, but is a "directive"
  ```
  @font-face {
    font-family: HandWriting;
    src: url('PAGSCapture.ttf');
  }
  ```
- Font family then available in rest of CSS
  ```
  p { font-family: HandWriting; ... }
  ```
- User agent dynamically downloads font
- Different syntaxes for font files
  - .ttf, .otf, .eot, .woff, .svg, ...
- Beware: copyright issues!
  - See [http://www.google.com/webfonts](http://www.google.com/webfonts)
Summary

- Colors: RGB vs HSL
- Images
  - Formats jpeg, png, gif, svg
  - Tradeoffs of size, quality, features
- Floating elements
  - Removed from flow, layered on top
- Fonts
  - Fallback fonts to account for uncertainty
  - Web fonts for dynamic loading