CSS Images Notes

1. Overview

- Focus: Controlling image sizes, alignment, and background images using CSS, separating presentation from HTML.
- Techniques:
  - Set dimensions with `width` and `height`.
  - Align with `float` or margins.
  - Use `background-image` for rollovers and sprites.

2. Controlling Image Sizes in CSS

- Use `width` and `height` (e.g., `img.large { width: 500px; height: 500px; }`).
- Benefits:
  - Smooth loading (browser reserves space).
  - Consistent sizes (e.g., e-commerce, grids).
- Example sizes: Small portrait (220×360px), small landscape (330×210px), feature photo (620×400px).
- Method:
  - Define size classes (`small`, `medium`, `large`) in CSS.
  - Apply via `class` (e.g., `<img src="images/magnolia-large.jpg" class="large" alt="Magnolia" />`).
- Example CSS:

```
img.large { width: 500px; height: 500px; }
img.medium { width: 250px; height: 250px; }
img.small { width: 100px; height: 100px; }
```

3. Aligning Images Using CSS

- Use `float` instead of `<img>` `align` attribute.
- Approaches:
  - Add `float` to size classes (e.g., `img.small { float: left; }`).
  - Use alignment classes (`align-left`, `align-right`) with size classes.
- Example:

```
<p><img src="images/magnolia-medium.jpg" alt="Magnolia" class="align-left
    ␣medium" />...</p>
```

```
img.align-left { float: left; margin: 0 10px 10px 0; }
img.align-right { float: right; margin: 0 0 10px 10px; }
img.medium { width: 250px; height: 250px; }
```

- Add `margin` to separate text from images.
- Applies to `<figure>` elements.

4. Centering Images Using CSS

- Convert inline images to block-level:

```
img.align-center { display: block; margin: 0 auto; }
```

- Alternatives:

  - `text-align: center` on container.
  - `margin-left: auto; margin-right: auto` on image.

- Example:

```
<p><img src="images/magnolia-medium.jpg" alt="Magnolia" class="align-
    center medium" />...</p>
```

5. Background Images

- **Adding** (`background-image`):

  - Apply to any element (e.g., `body { background-image: url("images/pattern.gif"); }`).
  - Repeats by default.

- **Repeating** (`background-repeat`):

  - Values: `repeat`, `repeat-x`, `repeat-y`, `no-repeat`.
  - Example: `background-repeat: repeat-x;`.

- **Position** (`background-position`):

  - Keywords (e.g., `center top`), pixels, percentages.
  - Example: `background-position: center top;`.

- **Shorthand** (`background`):

  - Combine: `background: color url("image") repeat position;`.
  - Example: `background: #ffffff url("images/tulip.gif") no-repeat top right;`.

- Note: Large images slow page load.

6. Image Rollovers & Sprites

- **Rollovers**:

  - Change style on hover/click using single image.
  - Shift `background-position` (e.g., `.button:hover { background-position: 0 -50px; }`).

- **Sprites**:

  - Single image for multiple elements (e.g., buttons, logos).
  - Reduces HTTP requests.

- Example:

```
<a class="button" id="add-to-basket">Add to basket</a>
```

```
.button { background-image: url("images/buttons.png"); background-
   position: 0 0; }
.button:hover { background-position: 0 -50px; }
```

7. **CSS3 Gradients**

- Linear gradients (e.g., `background-image:  linear-gradient(#336666, #66cccc);`).

- Use prefixes (`-webkit-`, `-moz-`, `-o-`) for compatibility.

- Fallback: `background-color` or `background-image`.

- Example:

```
.gradient {
    background-color: #66cccc;
    background-image: url("images/fallback-image.png");
    background-image: -webkit-linear-gradient(#336666, #66cccc);
    background-image: linear-gradient(#336666, #66cccc);
}
```

8. **Contrast of Background Images**

- **High Contrast**: Photos unsuitable (poor legibility).

- **Low Contrast**: Ideal; adjust in Photoshop/GIMP.

- **Screen**: Semi-transparent background (e.g., `rgba(0,0,0,0.5)`) behind text.

- Example: Low-contrast or screened backgrounds improve readability.

9. **Example Application**

- Image gallery:
  - *Background*: Repeating texture (`body { background-image:  url("images/backdrop.gif"); }`).
  - *Wrapper*: `width:  720px; margin:  0 auto;`.
  - *Layout*: Three columns (`.entry { width:  220px; float:  left; margin:  10px; }`).
  - *Figures*: `figure { width:  202px; height:  170px; background-color:  #e7e3d8; padding:  9px; }`.
  - *Images*: `figure img { width:  200px; height:  150px; border:  1px solid #d6d6d6; }`.
  - *Captions*: `figcaption { background-image:  url("images/icon.png"); background-rep no-repeat; padding-left:  20px; }`.
  - *Shadow*: `.entry { background-image:  url("images/shadow.png"); background-positi bottom; }`.

10. Summary

- Set image dimensions with CSS for consistency.

- Align with `float` (left/right) or `margin:  0 auto` (center).

- Use `background-image` for decorative effects.

- Create rollovers/sprites for interactivity and optimization.

- Ensure low-contrast or screened backgrounds for legibility.