CSS Summary & Review List

1. Core Concepts

- **Purpose**: CSS styles HTML elements (colors, layout, fonts)
- **How to apply CSS**:
  - Inline (`style` attribute)
  - Internal (`<style>` tag in `<head>`)
  - External (separate `.css` file linked with `<link>`)

2. Selectors (How to target elements)

- **Basic**:

```css
element { }        /* Tag (div, p, h1) */
.class { }         /* Class selector */
#id { }            /* ID selector (unique) */
* { }              /* Universal selector */
```

- **Combination**:

```css
.class1.class2 { } /* Multiple classes */
element.class { }  /* Element with class */
selector1, selector2 { } /* Group selectors */
```

3. Box Model (Every element is a box)

- **Components from inside out**:
  1. Content (text/images)
  2. Padding (space inside, around content)
  3. Border (edge around padding)
  4. Margin (space outside, between elements)
- **Box-sizing**:

```css
box-sizing: content-box; /* Default */
box-sizing: border-box; /* Width includes padding+border */
```

4. Common Properties

- Text:

```css
color, font-family, font-size, font-weight
text-align, text-decoration, line-height
```

- Colors:

```css
color: red;           /* Named */
color: #ff0000;       /* Hex */
color: rgb(255,0,0);  /* RGB */
color: rgba(255,0,0,0.5); /* RGB with transparency */
```

- Background:

```css
background-color, background-image
background-size, background-position
background: shorthand /* color image repeat position/size */
```

5. Display & Positioning

- Display types:

```css
display: block;    /* Full width, new line */
display: inline;   /* Flows with text, no width/height */
```

```
display: inline-block; /* Mix of both */
display: none;     /* Hidden and removed from layout */
```

- **Position**:

```css
position: static;    /* Default, normal flow */
position: relative;  /* Offset relative to itself */
position: absolute;  /* Relative to nearest positioned ancestor */
position: fixed;     /* Relative to viewport */
position: sticky;    /* Toggles between relative & fixed */
```

6. Layout Basics

- **Margin & Padding**:

```css
margin: 10px;                 /* All sides */
margin: 10px 20px;            /* Top/Bottom, Left/Right */
margin: 10px 20px 15px 5px;  /* Top, Right, Bottom, Left */
```

- **Flexbox** (for 1D layouts):

```css
display: flex;          /* Container */
justify-content: ;      /* Main axis alignment */
align-items: ;          /* Cross axis alignment */
flex-direction: ;       /* Row/column */
```

7. Specificity & Cascade

- **Order of importance**:

1. `!important` (avoid if possible)
2. Inline styles
3. ID selectors

   4. Class/attribute/pseudo-class selectors
   5. Element/pseudo-element selectors
- **Rule**: More specific selectors win over less specific

8. Pseudo-classes & Elements

- **Pseudo-classes** (state-based):

```css
:hover, :focus, :active
:first-child, :last-child, :nth-child(n)
```

- **Pseudo-elements** (styling parts):

```css
::before, ::after /* Add content */
::first-line, ::first-letter
```

9. Common Units

- **Absolute**: `px` (pixels)
- **Relative**:
  - `%` (percentage of parent)
  - `em` (relative to parent font-size)
  - `rem` (relative to root/html font-size)
  - `vw`, `vh` (viewport width/height percentage)

10. Must-Remember Tips

- CSS reads top-to-bottom (later rules override earlier)
- Use external CSS files for maintainability
- Test in multiple browsers
- Use developer tools (F12) to inspect/debug
- Start mobile-first for responsive design
- Comment your code: `/* This is a comment */`