

- **Resizing Images:** Use Photoshop for sharpening and adjustment before exporting.
- **Image Types:**
 - **JPG:** Ideal for photos and images with gradients.
 - **PNG-8:** Best for logos; supports transparency.
 - **GIF:** Useful for animation and simple transparent images.

Wireframe

- **Wireframes:** Blueprint for website layout.
- **CSS Types:**
 - **Inline:** Style applied directly on the element (`<h1 style="color: red">`).
 - **Embedded:** Style block inside `<head>` using `<style>`.
 - **External:** Link an external `.css` file using `<link>`.
- **Inheritance:** Styles cascade to child elements unless overridden.
- **DOM:** Document Object Model - Interface for modifying HTML/XML documents.
- **Relative Fonts:** %, em, rem improve screen adaptability.
- **CSS Structure:**
 - **Selector:** Element to style (p, .class).
 - **Property:** Defines the style (color, margin).
 - **Example:** `p { color: blue; }`

SITE MAPS, COLORS, & BOX MODEL

- **Development Stages:** Define, plan, design, develop, test, maintain.
- **Hex Color:** Colors using RGB

Black: #000000 -White: #FFFFFF -Red: #FF0000 -Blue: #0000FF -Green: #008000 -Yellow: #FFFF00

- **Pseudo-Selectors:** Target states like `:hover` or `:active`.
- **Descendant Selector:** Select child elements (e.g., `div p`).
- **Box Model:**
 - **Content:** Main area.
 - **Padding:** Space inside the border.
 - **Border:** Around padding.
 - **Margin:** Space outside the border.

CSS LAYOUT & FLEXBOX

- **Semantic Tags:** `<header>`, `<nav>`, `<main>`, `<footer>` for meaningful sections.
- **Div vs. Span:**
 - **Div:** Block-level, used for sections.
 - **Span:** Inline, used for small content.
- **CSS Float:** Floats content left/right.
- **What does it do?** When an element is floated, it's taken out of the normal document flow and positioned to the left or right edge of its containing block.
- **Clear:** Stops wrapping of floated elements.
- **Flexbox:**
 - `display: flex;` Enables flexbox.
 - `flex-direction:` Sets main axis (row/column).
 - `justify-content:` Aligns items (center, space-between).
 - `align-items:` Aligns items vertically.
 - `flex-wrap:` Wraps items to multiple lines.
 - **Example:** `.container {display: flex; justify-content: space-between; align-items: center;}`

CSS Flexbox Reference

Key Flexbox Properties

- **Flex Container Properties:**
 - **display: flex;** Turns on Flexbox for the container.
 - **flex-direction:** Sets the direction of flex items (row, column).
 - **justify-content:** Aligns items horizontally (center, space-around).
 - **align-items:** Aligns items vertically (stretch, flex-start).
 - **flex-wrap:** Allows items to wrap (wrap, nowrap).

Flexbox Item Properties **flex-grow:** Defines how much the item will grow.

- **flex-shrink:** Specifies how much the item will shrink.
- **flex-basis:** Sets the initial size of the item.
- **order:** Specifies the order of items.
- **align-self:** Overrides align-items for individual items.

Flexbox Example Code

```
.container {display: flex; flex-direction: row; justify-content: space-between; align-items: center; flex-wrap: wrap;}
.item1 {flex: 2;}
.item2 {flex: 1;}
```

Box Model Review

- **Content:** The actual content of the element.
- **Padding:** Space between content and border.
- **Border:** Surrounds the content.
- **Margin:** Space outside the border.

Exam Example

- **Question:** Use flexbox to make "Hello" take ¾ and "World" take ¼ of the space.

```
.welcome {display: flex;}
.hello {flex: 3;}
.world {flex: 1;}
```