

1. Overview

- CSS provides multiple properties to control **text appearance** on a webpage.
 - Key font-related properties:
 - color → text color
 - font-size → text size
 - font-weight → text thickness
 - font-family → typeface (style of font)
 - text-align → text alignment on the page
-

2. Font Size (font-size)

- Defines how large or small the text appears.
Example:
- `h1 { font-size: 20px; }`

a. Units for Font Size

Unit	Meaning	Relation	Notes
px	Pixels	1px = 1/96 inch ≈ 0.26mm	Static, screen-based
pt	Points	1pt = 1/72 inch ≈ 0.35mm	Used in print, e.g., Word uses points
em	Relative to parent element's font size	1em = 100% of parent	2em = 2 × parent size
rem	Relative to root (html) font size	1rem = 100% of root	More consistent for responsive design
Named sizes	e.g., small, large, x-large	Predefined by browser	Static and less precise

b. Example of em vs rem

```
html { font-size: 20px; }
```

```
body { font-size: 2em; } /* 40px if parent is 20px */
```

```
h2 { font-size: 2rem; } /* 40px regardless of parent */
```

- em → Scales with parent.

- rem → Scales with root (html), unaffected by nesting.
 - **Best practice:** Use **rem** for predictable scaling across nested elements.
-

3. Font Weight (font-weight)

- Controls **boldness** or **thickness** of text.

Value	Type	Example	Description
Keywords	normal, bold, bolder, lighter		Relative to default or parent
Numeric	100–900		400 = normal, 700 = bold, 900 = very bold

Example:

```
p { font-weight: 700; }
```

- lighter → decreases by 100
 - bolder → increases by 100
-

4. Font Family (font-family)

- Defines the **typeface** used for text.

Example:

```
h1 { font-family: Helvetica, sans-serif; }
```

a. Font Stacks

- Always provide a **fallback** (generic) typeface:
 - If primary font unavailable, browser uses fallback.
 - Generic types include:
 - serif → with small decorative “feet” (e.g., Times New Roman)
 - sans-serif → clean edges, no feet (e.g., Arial, Helvetica)
 - monospace → equal-width characters (code-like)
 - cursive → script-like, flowing letters
 - fantasy → decorative or stylized fonts

b. Multi-word Font Names

- Use quotes around font names with spaces:
- ```
p { font-family: "Times New Roman", serif; }
```

#### c. Custom Fonts (Google Fonts)

- Import using <link> tag in <head> from [fonts.google.com](https://fonts.google.com).
  - Example:
  - `<link href="https://fonts.googleapis.com/css2?family=Caveat&display=swap" rel="stylesheet">`
  - `h1 { font-family: "Caveat", cursive; }`
  - Ensures font loads correctly on all systems (Mac, Windows, etc.).
- 

## 5. Text Alignment (`text-align`)

- Controls horizontal positioning of text.

| Value | Description |
|-------|-------------|
|-------|-------------|

|      |                          |
|------|--------------------------|
| left | Aligns text to left edge |
|------|--------------------------|

|       |                           |
|-------|---------------------------|
| right | Aligns text to right edge |
|-------|---------------------------|

|        |                           |
|--------|---------------------------|
| center | Centers text horizontally |
|--------|---------------------------|

|       |                                                                        |
|-------|------------------------------------------------------------------------|
| start | Aligns based on writing direction (left for English, right for Arabic) |
|-------|------------------------------------------------------------------------|

|     |                   |
|-----|-------------------|
| end | Opposite of start |
|-----|-------------------|

Example:

```
h2 { text-align: center; }
```

---

## 6. Summary of Key Concepts

- font-size controls text size — use rem for consistency.
  - font-weight adjusts boldness (100–900).
  - font-family defines typeface — always include a fallback.
  - Use **Google Fonts** for universal, free font access.
  - text-align arranges text alignment on the screen.
  - Root-based sizing (rem) keeps font scaling predictable and manageable.
- 

### In short:

Mastering font properties in CSS lets you control *how your text feels* — from size and weight to style and alignment — ensuring your web pages look sharp, readable, and consistent everywhere.