

# CSS Font Properties : By Gagan Baghel

The `font` properties in CSS define the appearance of text, such as its size, style, weight, line height, and family. These properties give web developers control over how text looks on a webpage, ensuring readability, style, and consistency across browsers and devices.

## ▼ 1. `font-family`

The `font-family` property specifies the typeface to be used for text. Multiple fonts can be listed as fallbacks in case the primary font is unavailable.

### Syntax:

```
font-family: <family-name>, <generic-family>;
```

### Values:

- `<family-name>` : A specific font name (e.g., `Arial`, `Times New Roman`).
- `<generic-family>` : A generic classification of fonts (e.g., `serif`, `sans-serif`, `monospace`, `cursive`, `fantasy`).

### Example:

```
p {  
    font-family: "Arial", "Helvetica", sans-serif;  
}
```

### Notes:

- It is a good practice to list a generic family at the end to ensure compatibility across platforms.
- Custom web fonts can be loaded using `@font-face` or linked from a service like Google Fonts.

## ▼ 2. `font-size`

The `font-size` property sets the size of the text.

### Syntax:

```
font-size: <absolute-size> | <relative-size> | <length> | <percentage>;
```

### Values:

- `<absolute-size>`: Keywords like `xx-small`, `x-small`, `small`, `medium`, `large`, etc.
- `<relative-size>`: Keywords like `larger`, `smaller`, which are relative to the parent element's font size.
- `<length>`: A specific size in units like `px`, `em`, `rem`, `pt`, etc.
- `<percentage>`: A percentage of the parent element's font size (e.g., `120%`).

### Example:

```
h1 {  
    font-size: 2rem; /* relative to the root element */  
}  
  
p {  
    font-size: 16px; /* fixed size */  
}
```

### Notes:

- Use relative units (`em`, `rem`, or `%`) for responsive design.
- The default font size for browsers is typically 16px.

## ▼ 3. `font-style`

The `font-style` property specifies whether the text should be styled with normal, italic, or oblique font styles.

### Syntax:

```
font-style: normal | italic | oblique | initial | inherit;
```

### Values:

- `normal` : Default, no special style.
- `italic` : Italic style.
- `oblique` : Slanted style, often similar to italic but less formal.
- `initial` : Sets the property to its default value.
- `inherit` : Inherits the value from the parent element.

### Example:

```
em {  
    font-style: italic;  
}  
  
h2 {  
    font-style: oblique;  
}
```

### Notes:

- `italic` uses the actual italic version of the font if available, whereas `oblique` simply slants the text.
- Always check if a font supports `italic` and `oblique`.

## ▼ 4. `font-weight`

The `font-weight` property controls the thickness of the characters.

### Syntax:

```
font-weight: normal | bold | bolder | lighter | <number>;
```

### Values:

- `normal` : Standard weight (usually 400).

- `bold` : Bold text (usually 700).
- `bolder` : Bolder than the parent element's font weight.
- `lighter` : Lighter than the parent element's font weight.
- `<number>` : Specific weight values from `100` to `900`, where `100` is the lightest and `900` is the boldest.

### Example:

```
h1 {  
    font-weight: 700; /* Equivalent to bold */  
}  
  
p {  
    font-weight: 300; /* Lighter text */  
}
```

### Notes:

- Not all fonts support all numeric values.
- It's common to use `normal`, `bold`, or specific values like `400` and `700`.
- Using numeric values gives more granular control over the font thickness.

## ▼ 5. `font-variant`

The `font-variant` property allows for the use of small-caps or other alternative characters in fonts that support them.

### Syntax:

```
font-variant: normal | small-caps | all-small-caps | inherit;
```

### Values:

- `normal` : Default, no special variant.
- `small-caps` : Displays lowercase text as smaller uppercase letters.
- `all-small-caps` : Forces all text to display as small-caps.

- `inherit` : Inherits from the parent element.

#### Example:

```
p {  
    font-variant: small-caps;  
}
```

#### Notes:

- `small-caps` is often used in typographic design to create visual distinction for specific text.

## ▼ 6. `font-stretch`

The `font-stretch` property selects a normal, condensed, or expanded face from a font.

#### Syntax:

```
font-stretch: normal | ultra-condensed | extra-condensed | condensed | se  
mi-condensed | semi-expanded | expanded | extra-expanded | ultra-expa  
nded;
```

#### Values:

- `normal` : Default font width.
- `condensed` , `expanded` : Specifies narrower or wider versions of the font.
- Various degrees of width: `ultra-condensed` , `extra-condensed` , `semi-expanded` , etc.

#### Example:

```
h1 {  
    font-stretch: condensed;  
}
```

#### Notes:

- Not all fonts support `font-stretch` , so it may have no effect in some cases.

## ▼ 7. **line-height**

The `line-height` property sets the height between lines of text.

### Syntax:

```
line-height: normal | <number> | <length> | <percentage>;
```

### Values:

- `normal`: Default height, typically around 1.2 times the font size.
- `<number>`: A multiplier of the font size (e.g., 1.5 means 1.5 times the font size).
- `<length>`: A fixed length (e.g., 24px).
- `<percentage>`: A percentage of the font size (e.g., 150%).

### Example:

```
p {  
    line-height: 1.6;  
}
```

### Notes:

- Using relative values like 1.5 or percentages is preferable for responsive design.
- Larger `line-height` improves readability for longer paragraphs.

## ▼ 8. **font (Shorthand)**

The `font` property is a shorthand for setting multiple font-related properties in a single declaration.

### Syntax:

```
font: <font-style> <font-variant> <font-weight> <font-size>/<line-height>  
      <font-family>;
```

## Values:

- You can combine the values of the individual properties: `font-style`, `font-variant`, `font-weight`, `font-size`, `line-height`, and `font-family`.

## Example:

```
p {  
  font: italic small-caps bold 16px/1.5 "Arial", sans-serif;  
}
```

## Notes:

- The order of the values is important when using shorthand.
- At a minimum, you must specify `font-size` and `font-family`.

## ▼ 9. `@font-face`

The `@font-face` rule allows you to load custom fonts in your CSS.

### Syntax:

```
@font-face {  
  font-family: "MyFont";  
  src: url("myfont.woff2") format("woff2"),  
       url("myfont.woff") format("woff");  
}
```

## Example:

```
@font-face {  
  font-family: "OpenSans";  
  src: url("OpenSans-Regular.ttf") format("truetype");  
}  
  
body {  
  font-family: "OpenSans", sans-serif;  
}
```

### Notes:

- Ensure proper font formats are used for cross-browser compatibility (`woff`, `woff2`, `ttf`, etc.).
  - Use multiple formats in the `src` rule to support different browsers.
- 

### Summary of Key Points:

- `font-family` : Specifies the typeface or a list of fallbacks.
- `font-size` : Controls the size of the text, using either absolute or relative units.
- `font-style` : Sets the style of the font, such as normal, italic, or oblique.
- `font-weight` : Defines the thickness of the font.

Gagan Baghel