CSS Text Properties : By Gagan Baghel

The `text` properties in CSS allow you to control the appearance and layout of text content of the web pages.

These properties can manipulate aspects such as alignment, decoration, transformation, spacing, indentation, and shadow.

Below is a breakdown of the key `text`-related properties in CSS:

▼ 1. `Text-Color`

▼ 1. **Syntax**:

```
Selector {
  color: value;
}
```

- `selector`: Targets the HTML element whose text color you want to change.
- `value`: Specifies the color, which can be expressed in various formats
- we can pass the values in the multiple format like
  - Color Name
  - RGB value
  - RGBA Value
  - HSL value
  - HSLA Value
  - Hexadecimal

▼ Best Practices:

- **Consistency**: Maintain a consistent color palette throughout your website to create a cohesive design.

- **Accessibility**: Ensure that text colors are easy to read, especially for users with color blindness or low vision.
- **Maintainability**: Use CSS variables for managing colors across a website, making it easier to apply changes consistently.

Example using CSS variables:

```css
:root {
  --primary-color: #FF5733;
  --secondary-color: #2E8B57;
}

h1 {
  color: var(--primary-color);
}

p {
  color: var(--secondary-color);
}
```

This approach allows you to define and reuse color values across your stylesheets easily.

2. `text-align`

The `text-align` property sets the horizontal alignment of text within a block element.

Syntax:

```css
text-align: left | right | center | justify | start | end | match-parent;
```

Values:

- `left`: Aligns text to the left.
- `right`: Aligns text to the right.

- `center` : Centers the text.
- `justify` : Stretches the text so that each line has equal width.
- `start` : Aligns text based on the writing direction (left for LTR, right for RTL).
- `end` : Aligns text to the opposite side of `start` .
- `match-parent` : Inherits the alignment from the parent element.

Example:

```css
p {
  text-align: justify;
}
```

▼ 3. `text-decoration`

The `text-decoration` property adds decorative effects to text such as underlining, overlining, or strikethrough.

Syntax:

```css
text-decoration: none | underline | overline | line-through | blink;
```

Values:

- `none` : Removes any text decoration.
- `underline` : Adds a line below the text.
- `overline` : Adds a line above the text.
- `line-through` : Adds a line through the text (strikethrough).
- `blink` : Makes the text blink (no longer supported in modern browsers).

Example:

```css
a {
  text-decoration: none;
}
```

4. `text-transform`

The `text-transform` property controls the capitalization of text.

Syntax:

```
text-transform: none | capitalize | uppercase | lowercase;
```

Values:

- `none`: No transformation (default).
- `capitalize`: Capitalizes the first letter of each word.
- `uppercase`: Converts all characters to uppercase.
- `lowercase`: Converts all characters to lowercase.

Example:

```
h1 {
  text-transform: uppercase;
}
```

5. `text-indent`

The `text-indent` property specifies the indentation of the first line of a text block.

Syntax:

```
text-indent: <length> | <percentage>;
```

Values:

- `<length>`: Specifies a fixed indentation (e.g., `20px`, `1em`).
- `<percentage>`: Specifies indentation as a percentage of the element's width.

Example:

```
p {
  text-indent: 50px;
}
```

▼ 6. `letter-spacing`

The `letter-spacing` property controls the space between characters in text (also known as "tracking").

Syntax:

```
letter-spacing: normal | <length>;
```

Values:

- `normal`: Default spacing.
- `<length>`: Specifies extra spacing in units like `px`, `em`, etc.
- Negative values decrease the spacing.

Example:

```
p {
  letter-spacing: 0.1em;
}
```

▼ 7. `word-spacing`

The `word-spacing` property controls the space between words in a block of text.

Syntax:

```
word-spacing: normal | <length>;
```

Values:

- `normal`: Default word spacing.
- `<length>`: Specifies extra space between words.
- Negative values reduce the space between words.

Example:

```css
p {
  word-spacing: 10px;
}
```

▼ 8. `line-height`

The `line-height` property sets the amount of space between lines of text (also known as leading).

Syntax:

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

Values:

- `normal` : Default line spacing based on the font size. { 1.1 to 1.5 time of the font size based on the font family
- `<number>` : Multiplier for the font size (e.g., `1.5`).
- `<length>` : Specifies fixed space between lines (e.g., `20px`).
- `<percentage>` : Percentage of the font size (e.g., `150%`).
- if we pass negative values it takes it as a zero.

Example:

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

▼ 9. `text-shadow`

The `text-shadow` property adds shadows to text, enhancing the visual effect of the text.

Syntax:

```
text-shadow: <offset-x> <offset-y> <blur-radius> <color>;
```

Values:

- `<offset-x>`: Horizontal shadow offset.
- `<offset-y>`: Vertical shadow offset.
- `<blur-radius>`: (Optional) The blur radius of the shadow.
- `<color>`: The color of the shadow.

Example:

```
h1 {
  text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.5);
}
```

▼ 10. `white-space`

The `white-space` property controls how white space (such as spaces, tabs, and line breaks) is handled within the element.

Syntax:

```
white-space: normal | nowrap | pre | pre-wrap | pre-line
```

Values:

- `normal`: Collapses white space and wraps text as necessary.
- `nowrap`: Collapses white space but prevents text from wrapping to the next line.
- `pre`: Preserves white space and line breaks.
- `pre-wrap`: Preserves white space but wraps text when necessary.
- `pre-line`: Collapses white space but preserves line breaks.

Example:

```
pre {
  white-space: pre;
}
```

▼ 11. `text-overflow`

The `text-overflow` property determines how text that overflows the bounds of its container is handled.

Syntax:

```
text-overflow: clip | ellipsis | <string>
```

Values:

- `clip`: Clips the text without adding an ellipsis.
- `ellipsis`: Displays an ellipsis (`...`) to indicate overflow.
- `<string>`: A custom string to represent overflow.

Example:

```
div {
  text-overflow: ellipsis;
  white-space: nowrap;
  overflow: hidden;
}
```

▼ 12. `direction` and `unicode-bidi`

These properties are used together to control the text direction, especially for right-to-left languages.

Syntax:

```
direction: ltr | rtl;
unicode-bidi: normal | embed | bidi-override;
```

Example:

```
p {
  direction: rtl;
  unicode-bidi: bidi-override;
}
```

▼ Summary of Key Points:

- `text-align`: Aligns text horizontally within its container.
- `text-decoration`: Adds effects like underlining, overlining, or strikethrough.
- `text-transform`: Controls text capitalization.
- `text-indent`: Indents the first line of text.
- `letter-spacing`: Controls the space between characters.
- `word-spacing`: Controls the space between words.
- `line-height`: Sets the space between lines of text.
- `text-shadow`: Adds shadows to text.
- `white-space`: Controls how white space is handled.
- `text-overflow`: Determines how overflowing text is displayed.

These text-related CSS properties are fundamental for controlling and styling textual content, ensuring you can achieve the desired layout and presentation across various screen sizes and devices.