CSS Shadows

Explanation

CSS Shadows are used to add a shadow effect behind text or HTML elements.
They help in creating depth and make content stand out from the background.
Shadows improve the visual appearance of a webpage and give it a modern look.

Example (Text Shadow)

```
h1 {
  text-shadow: 2px 2px 5px gray;
}
```

Use:

Text shadow is used to highlight headings and important text.
It improves readability, especially when text is placed over images or colored backgrounds.

Shadow Effects

Explanation

Shadow effects control the appearance of shadows such as their position, blur, and color.
They allow designers to create soft or strong shadows depending on the design need.

Example

```
p {
  text-shadow: 1px 1px 3px black;
```

}

Use:
Shadow effects are used to make text clearer and more noticeable. They are commonly applied to titles, banners, and hero sections of websites.

Box Shadow

Explanation

Box shadow adds a shadow around elements like divs, buttons, and cards. It gives a 3D or raised effect, making elements look clickable or important.

Example

```
div {
  width: 250px;
  height: 120px;
  box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.4);
}
```

Use:
Box shadows are widely used in cards, buttons, forms, and containers. They help separate content sections and give a clean, professional UI design.

CSS Text Effects

CSS has some properties to handle text overflow, word wrapping, line breaking rules and writing modes.

In this chapter you will learn about the following properties:

- text-overflow - Specifies how to handle overflowed content
- word-wrap - Allows long words to be able to be broken and wrap onto the next line
- word-break - Specifies line breaking rules
- writing-mode - Specifies whether lines of text are laid out horizontally or vertically

---

CSS text-overflow Property

The CSS text-overflow property specifies how overflowed content that is not displayed should be signaled to the user. It can be clipped or rendered with ellipsis (...).

Both of the following properties are required for text-overflow to take effect:

- white-space: nowrap;
- overflow: hidden;

Here, the overflowed content is clipped:

This is some long text that will not fit in the box

Here, the overflowed content is rendered with ellipsis (...):

This is some long text that will not fit in the box

The CSS code is as follows:

Example

```
p.test1 {
  width: 200px;
  border: 1px solid #000000;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: clip;
}
```

```
p.test2 {
  width: 200px;
  border: 1px solid #000000;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
}
```

The following example shows how you can display the overflowed content when hovering over the element:

Example

```
p:hover {
  overflow: visible;
}
```

CSS word-wrap Property

The CSS word-wrap property allows long words to be able to be broken and wrap onto the next line.

The word-wrap property allows you to force the text to wrap - even if it means splitting it in the middle of a word:

The CSS code is as follows:

Example

Allow long words to be able to be broken and wrap onto the next line:

```
p {
  word-wrap: break-word;
}
```

CSS word-break Property

The CSS word-break property specifies how words should break when reaching the end of a line.

This property can take one of the following values:

- normal - This is default. Uses the default line breaking rules of the language
- break-all - Allows words to be broken at any character to prevent overflow
- keep-all - Prevents words from breaking

Here, we use normal:

This paragraph contains some text. This line will-break-at-hyphens.

Here, we use break-all:

This paragraph contains some text. The lines will break at any character.

The CSS code is as follows:

Example

```
p.test1 {
  word-break: normal;
}

p.test2 {
  word-break: break-all;
}
```

CSS writing-mode Property

The CSS writing-mode property specifies whether lines of text are laid out horizontally or vertically.

This property can take one of the following values:

- horizontal-tb - Default. The text flows horizontally from left to right, vertically from top to bottom
- vertical-rl - The text flows vertically from top to bottom, horizontally from right to left

- vertical-lr - The text flows vertically from top to bottom, horizontally from left to right

Here is a text with a span element with a vertical-rl writing-mode.

The following example shows some different writing modes:

Example

```
p.test1 {
  writing-mode: horizontal-tb;
}

span {
  writing-mode: vertical-rl;
}

p.test2 {
  writing-mode: vertical-rl;
}
```
