CSS Backgrounds

The CSS background properties are used to add background effects for elements.

1.CSS background-color

The `background-color` property specifies the background color of an element.

The background color of a page is set like this:

```
body {
  background-color: lightblue;
}
```

With CSS, a color is most often specified by:

- a valid color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

Other Elements

You can set the background color for any HTML elements:

```
div {
  background-color: lightblue;
}
```

2.Opacity / Transparency

The `opacity` property specifies the opacity/transparency of an element. It can take a value from 0.0 - 1.0. The lower value, the more transparent:

- `background-color: green;opacity: 0.1;`
- `background-color: green; opacity: 0.3;`
- `background-color: green; opacity: 0.6;`
- `background-color: green; opacity: 1;  //default`

Note: When using the `opacity` property to add transparency to the background of an element, all of its child elements inherit the same transparency. This can make the text inside a fully transparent element hard to read.

Transparency using RGBA

If you do not want to apply opacity to child elements,use **RGBA** color values.

you can use RGB as a color value. In addition to RGB, you can use an RGB color value with an **alpha** channel (RGB**A**) - which specifies the opacity for a color.

An RGBA color value is specified with: rgba(red, green, blue, *alpha*). The *alpha* parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque).

- background: rgb(0, 128, 0);
- background: rgba(0, 128, 0, 0.1);
- background: rgba(0, 128, 0, 0.3);
- background: rgba(0, 128, 0, 0.6);

3.CSS background-image

The `background-image` property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

Set the background image for a page:

```
body {
  background-image: url("paper.gif");
}
```

Background-repeat: no-repeat

Showing the background image only once is also specified by the `background-repeat` property:

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
}
```

4.CSS Gradients

CSS gradients let you display smooth transitions between two or more specified colors.

Linear Gradients

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

Syntax

```
background-image: linear-gradient(direction, color-stop1, color-stop2,
...);
```

Direction - Top to Bottom (this is default)

The following example shows a linear gradient that starts at the top. It starts red, transitioning to yellow:

- `background-image: linear-gradient(red, yellow);`

Direction

The first example shows a linear gradient that starts from the left. It starts red, transitioning to yellow:

- `background-image: linear-gradient(to right, red , yellow);`
- `background-image: linear-gradient(to left, red , yellow);`
- `background-image: linear-gradient(to top, red , yellow);`
- `background-image: linear-gradient(to bottom, red , yellow);`

Direction - Diagonal

You can make a gradient diagonally by specifying both the horizontal and vertical starting positions.

The following example shows a linear gradient that starts at top left (and goes to bottom right). It starts red, transitioning to yellow:

- `background-image: linear-gradient(to bottom right, red, yellow);`

Using Multiple Color Stops

The following example shows a linear gradient (from top to bottom) with multiple color stops:

```
#grad {
  background-image: linear-gradient(red, yellow, green);
}
```

The following example shows how to create a linear gradient (from left to right) with the color of the rainbow and some text:

```
#grad {
  background-image: linear-gradient(to right,
red,orange,yellow,green,blue,indigo,violet);
}
```

Using Transparency

CSS gradients also support transparency, which can be used to create fading effects.

To add transparency, we use the rgba() function to define the color stops. The last parameter in the rgba() function can be a value from 0 to 1, and it defines the transparency of the color: 0 indicates full transparency, 1 indicates full color (no transparency).

The following example shows a linear gradient that starts from the left. It starts fully transparent, transitioning to full color red:

- `background-image: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));`