Layering and compositing content with CSS

One of the things you can do with CSS is to layer content and have the layers perform different tasks.

Some of these techniques are inherited from, and used together with, SVG and others are specific to CSS.

CSS Backround Images

The first way I learned to layer content was using an image as the lowest item on the stack and then position text or other images on top of our 'base layer'.

```
<div class="back">
  <h1>Santiago, Chile</h1>
</div>
```

The background image has to have a size, in this case I chose to use viewport units to make sure that it'll work the same in all devices.

We absolutely position the H1 element so we can play with `top` and `left` to control the position on screen.

```
.back {
  background:
    url(https://s3-us-west-2.amazonaws.com/s.cdpn.io/32795/chile-01.jpg)
    no-repeat center center fixed;
  width: 100vw;
  height: 100vh;
  background-size: cover;
}

h1 {
  position: absolute;
```

```
  color: #fff;
  font-size: 6em;
  top: 50vh;
  left: 50vw;
}
```

With this technique you can also use CSS gradients as the background. The text remains the same.

```
<div class="back">
  <h1>Santiago, Chile</h1>
</div>
```

We modify the `.back` selector to use a linear gradient as the background.

```
.back {
  background: red;
  background:
    linear-gradient(
      to bottom right,
      white, rebeccapurple);
  height: 100vh;
  width: 100vw;
  background-size: cover;
}

h1 {
  color: #fff;
  position: absolute;
  left: 50%;
  top: 45vh;
}
```

CSS Tricks explains techniques and best practices in CSS Gradients so I'll defer to them in terms of what you can do with gradients.

Z-Index

The next way to layer content is to use z-index. The idea is that elements with the higher z-index value are laid “closer to the user” (over other elements) while those with lower or negative are laid “away from the user” (below elements with higher z-index).

The example below uses an image element for the background instead of relying on CSS background images.

```
<img class="layer1"
  src="https://s3-us-west-2.amazonaws.com/s.cdpn.io/32795/santiago002.jpg"
<h1>Santiago, Chile</h1>
```

Because we’re using an element that is in the DOM we need to position the base layer image relative too the other content for this technique to work.

The h1 element is positioned absolutely and placed usng a combnination of `top`, `left` and `z-index` to place it on screen relative to the background image.

```
.layer1 {
  position: relative;
  width: 100%;
  height: auto;
}

h1 {
  color: #fff;
  font-size: 6rem;
  position: absolute;
  z-index: 1;
  top: 1.5em;
  left: 40%;
}
```

blending and composition

There are times when we want more thaan we can do owith just layering. It wasn't until the Compositing and Blending Level 1 specification there was no equivalent way to do blending with CSS other than opacity.

The spec defines two modes to blend content: `background-blend-mode` and `mix-blend-mode`

background-blend-mode

This mode allows you to blend elements (images or colors) with their background. It also allows for multiple background images and blend modes so different portionos of the image will blend with their background differently.

An example of using `background-blend-mode` with one background image and a single color looks like this:

```
.back {
  background:
    url(https://s3-us-west-2.amazonaws.com/s.cdpn.io/32795/chile-02.jpg)
    center center;
  background-color: orange;
  background-blend-mode: hard-light;
  height: 100vh;
  width: 100vw;
}
```

We can add more background images and background blend modes to create more complex effects.

mix-blend-mode

`Mix-blend-mode` lets us use things other than images in the blending. I love to use this for blending text with a color and a background image.

The example uses a single image, a background color defined in CSS and a string.

```
<div>
  <img src="https://s3-us-west-2.amazonaws.com/s.cdpn.io/32795/chile-01.jp
  <h1>Santiago, Chile</h1>
</div>
```

We give the `div` element a relative position so we can layer content abooove it and have elements “higher” in the stacking oroder blend with it.

We use an rgba color to the background color so we can control its opacity.

We position it absolutely so we can move it aruond by controlling the top and left attributes.

We make the full string uppercase with `text-transform` and use a `overlay` blend mode.

```
div {
  position: relative;
}

img {
  width: 100%;
}

h1 {
  background-color: rgba(201,201,201,.01);
  padding: .5em 1em;
  position: absolute;
  top: 0;
  left: 20vw;
  width: 75%;
  font-size: 10vw;
  font-weight: 900;
  text-transform: uppercase;
  mix-blend-mode: overlay;
}
```

In the example you can see how only the text inherits the color of the background

image at the same position. The rest of the image looks a little hazy but, to me, that enhances the effect.

Another example, taken from Exploring Blend Modes in CSS blends two images together.

The HTML code has the twoo images inside a `div`.

```
<div class="blend1">
  <img  src="/images/monkey.jpg"
        width="400" height="600">
  <img  src="/images/sky.jpg"
        width="400" height="600">
</div>
```

The CSS changes the position of the first image and uses a soft-light `mix-blend-mode`.

```
.blend1 img:first-child {
  position: absolute;
  mix-blend-mode: soft-light;
}
```

Isolate

Setting isolation to isolate will turn the element into a stacking context, and control if the element's contents can blend with their backdrop outside this context.

By default, the isolation property is set to auto – which implies that they are not isolated.

This feature gives us tighter control over blending as we can choose which elements of our stack we blend and which one to leave out of the blending process.

The blend modes

There are 16 blend modes available in CSS and they will each change the way the

blended images look.

- `normal` The default blend mode and indicates that no blending is applied
- `multiply`
- `screen`
- `overlay`
- `darken`
- `lighten`
- `color-dodge`
- `color-burn`
- `hard-light`
- `soft-light`
- `difference`
- `exclusion`
- `hue`
- `saturation`
- `color`
- `luminosity`

Sara Soedain's Compositing And Blending In CSS has explanations of what these modes do.

Links and Resources

- Compositing and Blending Level 1 specification
- Compositing and Blending Level 2 editor's draft
- Porter Duff compositing operations
- Using SVG with CSS3 and HTML5 – O'Reilly Media
- Using SVG with CSS3 and HTML5 – Playing with Pixels: Filters and Blend Modes
- Compositing And Blending In CSS — Sara Soueidan
- Exploring Blend Modes in CSS — Alligator.io