

CSS Miscellaneous Properties

1. CSS Specificity:

- a. Specificity is a set of rules that CSS uses to determine which style declaration takes priority when multiple rules target the same element.
- b. Why is it important?
- c. When multiple CSS rules apply to the same element, *specificity decides which one is applied.*
- d. How is specificity calculated?
- e. Inline styles (`style=""`) → 1000
- f. ID selectors (`#id`) → 100
- g. Class, attribute, and pseudo-class selectors (`.class`, `[type]`, `:hover`) → 10
- h. Element and pseudo-element selectors (`div`, `h1`, `::before`) → 1
- i. Example ⇒

```
<style>
  /* specificity = 1 */
  h1 {
    color: ■red;
  }
  /* specificity = 10 */
  .heading {
    color: ■blue;
  }
  /* specificity = 100 */
  #title {
    color: ■green;
  }
</style>
</head>
<body>
  <h1 id="title" class="heading">Hello</h1>
</body>
```

2. CSS opacity Property:

- a. The **opacity** property in CSS controls the transparency level of an element.

- b. **Syntax:**

```
selector {  
    opacity: value;  
}
```

- c. Example ⇒

```
.box {  
    opacity: 0.5; /* 50% transparent */  
}
```

3. CSS object-fit Property:

- a. The **object-fit** property defines how an **** or **<video>** should be resized to fit its container.

- b. To control the resizing behavior of images/videos inside a container with fixed dimensions.

- c. **Syntax:**

```
img {  
    object-fit: cover | contain | fill | none | scale-down;  
}
```

- d. **Values:**

- i. **fill** (default): Stretches the media to fill the box
- ii. **contain**: Resizes to fit inside the box without cropping

- iii. **cover**: Resizes to cover the box completely, cropping if necessary
- iv. **none**: Original size is preserved
- v. **scale-down**: Chooses the smaller of **none** or **contain**

4. CSS filter Property:

- a. The **filter** property applies visual effects (like blur, brightness, contrast) to elements, typically images.
- b. To apply **graphic effects** to elements without editing the image itself.

c. **Syntax:**

```
selector {
    filter: value;
}
```

d. You can apply multiple filters:

```
.box {
    filter: grayscale(50%) brightness(120%);
}
```

e. **values:**

```
filter : none;

blur() -> px
brightness() -> %
contrast() -> %
drop-shadow() -> h-shadow v-shadow blur sp

read color;

grayscale() -> %
hue-roatate() -> deg
invert() -> %
opacity() -> %
saturate() -> %
url()
```