

DIV , Layers , Filters in CSS

<div> Tag

- The <div> tag defines a division or a section in an HTML document.
- The <div> element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript.
- The <div> element is very often used together with CSS, to layout a web page.
- By default, browsers always place a line break before and after the <div> element. However, this can be changed with CSS.
- **Syntax:**
`<div> Contents and other tags </div>`

DIV TAG

Example

A section in a document that will have a light blue background color:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>This is some text.</p>
```

```
<div style="background-color:lightblue">
```

```
  <h3>This is a heading in a div element</h3>
```

```
  <p>This is some text in a div element.</p>
```

```
</div>
```

```
<p>This is some text.</p>
```

```
</body>
```

```
</html>
```

 tag

- The HTML tag is used for grouping and applying styles to inline elements.
- There is a difference between the span tag and the div tag. The span tag is used with inline elements i.e. style parts of the text while the div tag is used with block-level content.
- **Syntax:**
` Contents and other tags `

- **Example:**

```
<html>
```

```
<head><title> Span Tag </title></head>
```

```
<body>
```

```
  <p>This is a paragraph <span style = "color:red;">
```

```
    This is a paragraph</span>This is a paragraph</p>
```

```
  <p><span style = "color:blue;">
```

```
    This is another paragraph</span></p>
```

```
</body>
```

```
</html>
```

Position Property

- The position property specifies the type of positioning method used for an element .
 - **Position:static** - Default value. Elements render in order, as they appear in the document flow. Top, left, right, bottom properties will not work.
 - **Position: fixed** – Elements with position fixed always stays in the same place even if the page is scrolled. Top, left, right, bottom properties will work.
 - **Position: relative** - The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position.
 - **Position: absolute** - The element is positioned relative to its first positioned (not static) ancestor element.
- Elements are then positioned using the top, bottom, left and right properties.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid green;
}

div.absolute {
  position: absolute;
  top: 80px;
  right: 0;
  width: 200px;
  height: 100px;
  border: 3px solid green;
}
</style>
</head>
<body>
<h2>position: absolute;</h2>
<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>
</body>
</html>
```

CSS-Layers

- CSS gives you opportunity to create layers of various divisions. The CSS layers refer to applying the *z-index* property to elements that overlap with each other.
- The *z-index* property is used along with the *position* property to create an effect of layers. You can specify which element should come on top and which element should come at bottom.
- A *z-index* property can help you to create more complex webpage layouts.

z-index property

- When elements are positioned, they can overlap other elements.
- The z-index property specifies the stack order of an element.
- An element with greater stack order is always in front of an element with a lower stack order.
- If two positioned elements overlap without a z-index specified, the element positioned last in html code will be shown on top.
- z-index only works on positioned elements (position:absolute, position:relative, or position:fixed).
- **Syntax:**

z-index:number;

Example

```
<html>
  <head>
  </head>
  <body>
    <div style="background-color:red; width:300px; height:100px;
position:relative; top:10px; left:80px; z-index:2">
      </div>

    <div style="background-color:yellow; width:300px; height:100px;
position:relative; top:-60px; left:35px; z-index:1;">
      </div>

    <div style="background-color:green; width:300px; height:100px;
position:relative; top:-220px; left:120px; z-index:3;">
      </div>
  </body>
</html>
```

Filter Effects

- You can use CSS filters to add special effects to text, images and other aspects of a webpage.
- The filter property is not supported by all browsers.

CSS Syntax

filter: none | blur() | brightness() | contrast() | drop-shadow() | grayscale() | hue-rotate() | invert() | opacity() | saturate() | sepia() ;

- To use multiple filters, separate each filter with a space

Filter Effects

- **none** : Default value. Specifies no effects.
- **blur(px)**: Applies a blur effect to the image. A larger value will create more blur. If no value is specified, 0 is used.
- **brightness(%)**: Adjusts the brightness of the image. 0% will make the image completely black. 100% (1) is default and represents the original image. Values over 100% will provide brighter results.
- **contrast(%)**: Adjusts the contrast of the image. 0% will make the image completely black. 100% (1) is default and represents the original image. Values over 100% will provide results with less contrast.
- **grayscale(%)**: Converts the image to grayscale. 0% (0) is default and represents the original image. 100% will make the image completely gray (used for black and white images).
- **hue-rotate(deg)**: Applies a hue rotation on the image. The value defines the number of degrees around the color circle the image samples will be adjusted. 0deg is default, and represents the original image.
Note: Maximum value is 360deg.

Filter Effects

- **invert(%):** Inverts the samples in the image. 0% (0) is default and represents the original image. 100% will make the image completely inverted.
- **opacity(%):** Sets the opacity level for the image. The opacity-level describes the transparency-level. 0% is completely transparent. 100% (1) is default and represents the original image (no transparency).
- **saturate(%):** Saturates the image. 0% (0) will make the image completely un-saturated. 100% is default and represents the original image. Values over 100% provides super-saturated results.
- **sepia(%):** Converts the image to sepia. 0% (0) is default and represents the original image. 100% will make the image completely sepia.
- **drop-shadow(*h-shadow v-shadow blur spread color*):** Applies a drop shadow effect to the image.

Filter Effects

Possible values:

h-shadow - Required. Specifies a pixel value for the horizontal shadow. Negative values place the shadow to the left of the image.

v-shadow - Required. Specifies a pixel value for the vertical shadow. Negative values place the shadow above the image.

blur - Optional. This is the third value, and must be in pixels. Adds a blur effect to the shadow. A larger value will create more blur (the shadow becomes bigger and lighter). Negative values are not allowed. If no value is specified, 0 is used (the shadow's edge is sharp).

spread - Optional. This is the fourth value, and must be in pixels. Positive values will cause the shadow to expand and grow bigger, and negative values will cause the shadow to shrink. If not specified, it will be 0 (the shadow will be the same size as the element).

Note: Chrome, Safari and Opera, and maybe other browsers, do not support this 4th length; it will not render if added.

color - Optional. Adds a color to the shadow. If not specified, the color depends on the browser (often black).

An **example** of creating a red shadow, which is 8px big both horizontally and vertically, with a blur effect of 10px:

filter: drop-shadow(8px 8px 10px red);