**Position:-** The position property specifies the type of positioning method used for an element. There are five different position values. Elements are then positioned using the top, bottom, left, and right properties.

* static:- position: static; it is always positioned according to the normal flow of the page: HTML elements are positioned static by default. Static positioned elements are not affected by the top, bottom, left, and right properties.
* relative:- the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.
* Fixed:- position: fixed; positions an element relative to the viewport, which means it always stays in the same place even if the page is scrolled. A fixed element does not leave a gap in the page where it would normally have been located.
* absolute:- An element with position: absolute; is positioned relative to the nearest positioned ancestor. If an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.
* Sticky:- An element with position: sticky; is positioned based on the user's scroll position. A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place

**Positioning Text In an Image:-**

e.g. <!DOCTYPE html>

<html>

<head>

<style>

.container {

position: relative;

}

.topleft {

position: absolute;

top: 8px;

left: 16px;

font-size: 18px;

}

img {

width: 100%;

height: auto;

opacity: 0.3;

}

</style>

</head>

<body>

<h2>Image Text</h2>

<p>Add some text to an image in the top left corner:</p>

<div class="container">

<img src="img\_5terre\_wide.jpg" alt="Cinque Terre" width="1000" height="300">

<div class="topleft">Top Left</div>

</div>

</body>

</html>

**Z-index:-** The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

An element can have a positive or negative stack order:

e.g. img {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}

**Overflow:-**The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.

The overflow property has the following values:

* visible - Default. The overflow is not clipped. The content renders outside the element's box
* hidden - The overflow is clipped, and the rest of the content will be invisible
* scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content
* auto - Similar to scroll, but it adds scrollbars only when necessary
* overflow-x specifies what to do with the left/right edges of the content.
* overflow-y specifies what to do with the top/bottom edges of the content.

e.g. div {  
  width: 200px;  
  height: 65px;  
  background-color: coral;  
  overflow: visible;  
}

**Styling Images:-**

**Rounded images:-** To create rounded images use border-radius property.

If border-radius:8px; is used then image will be displayed with rounded corners.

If border-radius:50%; is used then image will be displayed in circle.

**Thumbnail images:-** Use border property to create thumbnail images.

e.g. img {  
  border: 1px solid #ddd;  
  border-radius: 4px;  
  padding: 5px;  
  width: 150px;  
}

**Thumbnail image as hyperlink:-**

e.g. img {  
  border: 1px solid #ddd;  
  border-radius: 4px;  
  padding: 5px;  
  width: 150px;  
}  
  
img:hover {  
  box-shadow: 0 0 2px 1px rgb(0, 140, 186,);  
}  
  
<a href="abc.html">  
  <img src="abc.jpg" alt="Paris">  
</a>

**Transparent image:-**

img

{  
  opacity: 0.5;  
}

**Image Text:-** Text can be inserted on image by using following properties.

.a {

position: absolute;

top: 50%;

left: 50%;

font-size: 18px;

}

**Image filters:-** The CSS filter property adds visual effects (like blur and saturation) to an element.

img {  
  filter: grayscale(100%);  
}

.blur {filter: blur(4px);}

.brightness {filter: brightness(250%);}

.contrast {filter: contrast(180%);}

.grayscale {filter: grayscale(100%);}

.huerotate {filter: hue-rotate(180deg);}

.invert {filter: invert(100%);}

.opacity {filter: opacity(50%);}

.saturate {filter: saturate(7);}

.sepia {filter: sepia(100%);}

.shadow {filter: drop-shadow(8px 8px 10px green);}

**Flip an image:-**

img:hover

{  
  transform: scaleX(-1);  
}

**Image reflection:-** The box-reflect property is used to create an image reflection.

The value of the box-reflect property can be: below, above, left, or right.

img {  
  -webkit-box-reflect: below;  
}

**Image reflection offset:-**

To specify the gap between the image and the reflection, add the size of the gap to the box-reflect property.

img

{  
  -webkit-box-reflect: below 20px;  
}