**CSS PROPERTIES**

|  |  |
| --- | --- |
| **Properties** | **Usage** |
| box-sizing : border-box | By default, in the CSS box model, the width and height you assign to an element is applied only to the element's content box. If the element has any border or padding, this is then added to the width and height to arrive at the size of the box that's rendered on the screen. This means that when you set width and height, you have to adjust the value you give to allow for any border or padding that may be added. |
| text-align: center | This property helps the element content to move to center. For void Elements like img tag we should wrap img around any non-semantic element like div and on the div we apply this property. |
| background-color: yellow | To apply the color fully/completely on the element we can use property. |
| background: linear-gradient(0deg, rgb(227, 255, 253),rgb(202,243,240) ); | To apply background color with gradient, I mean gradually changing from one angle to another angle. We use linear-gradient function which takes three arguments. First one is degree of angle and second one is starting color, last one is ending color. |
| box-shadow: 2px 4px 8px rgb(68, 67, 67); | This gives a shadow to element. 2px left to right, 4px top to bottom, 8px blur radius and color of shadow will be declared in rgb. |
| object-fit: cover; | This property is applied to image to get most of its aspect ratio back. This property is applied when we declared width and height for images then aspect ratio or originality of image is disturbed. |
| background-image: url("/images/places/ocean.jpg"); | To apply an image to background. |
| background-position: center;  background-size: cover; | These properties go around with background-image property. background-size: cover is used to spread 100% image correctly to screen. |
| position: relative | When add this property to any element, that property is free to move top, bottom, left, right with respective to its previous element. Default position is static. |
| position: absolute | The element is removed from the normal document flow, and no space is created for the element in the page layout. It is positioned relative to its closest positioned ancestor, if any; otherwise, it is placed relative to the initial containing block. Its final position is determined by the values of top, right, bottom, and left. |
| text-decoration: none | This one is used on anchor tag to get rid of underline. |
| list-style: none | This one is used on ul or li tags to get rid of default bullets or numbers on list items. |
| border: 5px solid rgb(98,98,98) | This one is used define a border around any element with border width and type of border and border color. |
| border-radius: 5px | This one is used on elements to get rounded borders. |
| text-transform: uppercase | This one is used to transform the text to uppercase or lowercase. |
| text-shadow: 1px 1px 2px rgb(0,0,0) | This gives a shadow to text. 1px left to right, 1px top to bottom, 2px blur radius and color of shadow will be declared in rgb. |
| display: flex | This one is used on a container element, where its children will be affected. |
| flex-direction: row | Block elements will flow as inline elements in a row. |
| flex-wrap: nowrap | This one doesn’t wrap the elements by default |
| justify-content: space-between | This one is used to manage the space with flex items. |
| display: grid | This one is used to make the child items as grid elements and 2d layout is possible. |
| grid-template-columns: 1fr 1fr; | This one is used to define how many columns in a grid and to say columns width in grid. |
| gap: 20px 20px | This one is used to define row gap and column gap in grid box. |
| main li: nth-of-type(3) {  grid-column: 1 / 3;  } | This one is used to define a third element in a grid list to occupy the column1 and column 2. |
| display: inline-block | For Inline elements we can’t add top and bottom margin and top and bottom padding and element won’t push the next line elements top or down, to resolve this problem we can use property display: inline-block. But there is an exception on replaced inline elements like img tag. |
| transform: scale(2) | Transformation : Move / change appearance of element, e.g. when hovering. |
| transition: transform 0.5s ease-out 1s; | Smooth transition from initial state to transformed state. It is applied to “initial state” of the element, not on event triggering the transition. |
| Font: normal 1rem ‘Roboto’, sans-serif | This one is used to define all together font-style, font-style, font-family.  font-style: normal, font-size: 1rem, font-family: Roboto |
| font: inherit | This inherit can be applied on many elements. If you have provided a font-family: sans-serif property in html element. And for button element you haven’t provided any font-family in CSS file but browser defaults will kick in for button element and provide font-family: Arial to it. But we want sans-serif which is applied to html element in that case we can declare button element and inside we can provide font-family: inherit. |