|  |  |  |
| --- | --- | --- |
| **EX.NO-3(i)** | **CASCADING STYLE SHEET** | **KAMALI.A**  **22Z436**  **B.E-CSE(G1)** |

**CSS INTRODUCTION:**

* CSS stands for Cascading Style Sheets.
* CSS saves a lot of work. It can control the layout of multiple web pages all at once.
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media
* CSS saves a lot of work. It can control the layout of multiple web pages all at once
* External stylesheets are stored in CSS files
* Cascading Style Sheets (CSS) is used to format the layout of a webpage.
* With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

**CSS COMMENTS:**

* CSS comments are not displayed in the browser, but they can help document your source code.
* CSS Comments are used to explain the code, and may help when you edit the source code at a later date.
* A CSS comment is placed inside the <style> element, and starts with /\* and ends with \*/.

|  |  |
| --- | --- |
| Example | Output |
| <html>  <style>  /\* This is a single-line comment \*/  p { color:black; }  </style>  <body>  <p>Kamali</p>  <p>22z436</p>  </body>  </html> |  |

**CSS BACKGROUNDS:**

* The CSS background properties are used to add background effects for elements.
* In these chapters, you will learn about the following CSS background properties:
* background-color
* background-image
* background-repeat
* background-attachment
* background-position
* background (shorthand property)
* CSS background-color
* The background-color property sets the background color of an element.
* The background of an element is the total size of the element, including padding and border (but not the margin).
* Syntax

|  |
| --- |
| background-color: color|transparent|initial|inherit; |

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  body {  background-color: lightblue;}  </style><body>  <h1>HTML</h1>  <p>hypertext markup language</p>  </body></html> |  |

* CSS background-image
* The background-image property specifies an image to use as the background of an element.
* By default, a background-image is placed at the top-left corner of an element, and repeated both vertically and horizontally.
* Syntax

|  |
| --- |
| background-image: url|none|initial|inherit; |

|  |  |
| --- | --- |
| Example | Output |
| <html>  body {  background-image: url("paper.gif"); }  <body>  <h1>Kamali.A</h1>  </body></html> |  |

* This example shows a bad combination of text and background image. The text is hardly readable

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  Body{  background-image: url("s.jpeg"); }  </style><body>  <p>KamaliArumugam</p>  </body>  </html> |  |

* The background image can also be set for specific elements, like the <p> element:

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  p {  background-image: url("paper.gif"); }  </style><body>  <h1>KAMALI</h1>  <p>22Z436</p>  </body></html> |  |

**CSS BORDERS:**

* The CSS border properties allow you to specify the style, width, and color of an element's border.
* CSS Border Style
* The border-style property specifies what kind of border to display.
* The following values are allowed:
* dotted - Defines a dotted border
* dashed - Defines a dashed border
* solid - Defines a solid border
* double - Defines a double border
* groove - Defines a 3D grooved border. The effect depends on the border-color value
* ridge - Defines a 3D ridged border. The effect depends on the border-color value
* inset - Defines a 3D inset border. The effect depends on the border-color value
* outset - Defines a 3D outset border. The effect depends on the border-color value
* none - Defines no border
* hidden - Defines a hidden border
* The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

|  |  |
| --- | --- |
| Example | Output |
| <html>  <style>  p.dotted {border-style: dotted;}  p.dashed {border-style: dashed;}  p.solid {border-style: solid;}  p.double {border-style: double;}  p.groove {border-style: groove;}  p.ridge {border-style: ridge;}  p.inset {border-style: inset;}  p.outset {border-style: outset;}  p.none {border-style: none;}  p.hidden {border-style: hidden;}  p.mix {border-style: dotted dashed soliddouble;}  </style><body>  <p class="dotted">A dotted border.</p>  <p class="dashed">A dashed border.</p>  <p class="solid">A solid border.</p>  <p class="double">A double border.</p>  <p class="groove">A groove border.</p>  <p class="ridge">A ridge border.</p>  <p class="inset">An inset border.</p>  <p class="outset">An outset border.</p>  <p class="none">No border.</p>  <p class="hidden">A hidden border.</p>  <p class="mix">A mixed border.</p>  </body>  </html> |  |

**CSS MARGINS:**

* Margins are used to create space around elements, outside of any defined borders.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  div {  margin: 30px;  border: 3px solid #4CAF50;  }  </style><body>  <h2>CSS Margins</h2>  <div>HTML</div>  </body></html> |  |

* The CSS margin properties are used to create space around elements, outside of any defined borders.
* With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).
* Margin - Individual Sides
* CSS has properties for specifying the margin for each side of an element:
* margin-top
* margin-right
* margin-bottom
* margin-left
* All the margin properties can have the following values
* auto - the browser calculates the margin
* length - specifies a margin in px, pt, cm, etc.
* % - specifies a margin in % of the width of the containing element
* inherit - specifies that the margin should be inherited from the parent element

**CSS PADDING:**

* Padding is used to create space around an element's content, inside of any defined borders.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  div {  padding: 30px;  border: 1px solid #4CAF50; }  </style><body>  <h2>CSS Padding</h2>  <div>kamali</div>  </body></html> |  |

* The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
* With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).
* Padding - Individual Sides
* CSS has properties for specifying the padding for each side of an element:
* padding-top
* padding-right
* padding-bottom
* padding-left
* All the padding properties can have the following values:
* length - specifies a padding in px, pt, cm, etc.
* % - specifies a padding in % of the width of the containing element
* inherit - specifies that the padding should be inherited from the parent element
* Example

|  |  |
| --- | --- |
| Example | Output |
| <html>  <style>  div {  border: 1px solid black;  background-color: lightblue;  padding-top: 50px;  padding-right: 30px;  padding-bottom: 50px;  padding-left: 80px; }  </style><body>  <h2> padding </h2>  <div>Kamali</div>  </body></html> |  |

**CSS HEIGHT, WIDTH :**

* The CSS height and width properties are used to set the height and width of an element.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  div {  height: 50px;  width: 30%;  border: 1px solid #4CAF50;}  </style><body>  <h2>CSS height and width</h2>  <div>Kamali</div>  </body>  </html> |  |

* The height and width properties are used to set the height and width of an element.
* The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.
* CSS height and width Values
* The height and width properties may have the following values:
* auto - This is default. The browser calculates the height and width
* length - Defines the height/width in px, cm, etc.
* % - Defines the height/width in percent of the containing block
* initial - Sets the height/width to its default value
* inherit - The height/width will be inherited from its parent value

**CSS BOX MODEL:**

* In CSS, the term "box model" is used when talking about design and layout.
* The CSS box model is essentially a box that wraps around every HTML element. It consists of: content, padding, borders and margins. The image below illustrates the box model:
* Explanation of the different parts:
* Content - The content of the box, where text and images appear
* Padding - Clears an area around the content. The padding is transparent
* Border - A border that goes around the padding and content
* Margin - Clears an area outside the border. The margin is transparent

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  div {  background-color: lightgrey;  width: 100px;  border: 15px solid green;  padding: 50px;  margin: 20px;}  </style><body>  <h2>Box Model</h2>  <div>KAMALI</div>  </body></html> |  |

**CSS OUTLINE:**

* An outline is a line drawn outside the element's border.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  p {  border: 2px solid black;  outline: #4CAF50 solid 10px;  margin: auto;  padding: 20px;  text-align: center;}  </style><body>  <h2>CSS Outline</h2>  <p>HTML</p>  </body></html> |  |

* CSS has the following outline properties:
* outline-style
* outline-color
* outline-width
* outline-offset
* outline
* The outline-style property specifies the style of the outline, and can have one of the following values:
* dotted - Defines a dotted outline
* dashed - Defines a dashed outline
* solid - Defines a solid outline
* double - Defines a double outline
* groove - Defines a 3D grooved outline
* ridge - Defines a 3D ridged outline
* inset - Defines a 3D inset outline
* outset - Defines a 3D outset outline
* none - Defines no outline
* hidden - Defines a hidden outline

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  p {outline-color:red;}  p.dotted {outline-style: dotted;}  p.dashed {outline-style: dashed;}  p.solid {outline-style: solid;}  p.double {outline-style: double;}  </style><body>  <h2>The outline-style Property</h2>  <p class="dotted">A dotted outline</p>  <p class="dashed">A dashed outline</p>  <p class="solid">A solid outline</p>  <p class="double">A double outline</p>  </body></html> |  |

**CSS TEXT:**

* CSS has a lot of properties for formatting text.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  div {  border: 1px solid gray;  padding: 5px;}  h3 {  text-align: center;  text-transform: uppercase;  color: #4CAF50;}  p {  text-indent: 20px;  text-align: justify;  letter-spacing: 2px;}  a {  text-decoration: none;  color: #008CBA;}  </style><body>  <div>  <h3>text formatting</h3>  <p>Kamali,my roll\_no  <a target="\_blank" href="tryit.asp?filename=trycss\_text">"22z436"</a> link.</p>  </div></body></html> |  |

* CSS contain
* Text Color
* Text Alignment
* Text Decoration
* Text Transformation
* Text Spacing
* Text Shadow
* Text Color
* The color property is used to set the color of the text.
* The color is specified by:
* a color name - like "red"
* a HEX value - like "#ff0000"
* an RGB value - like "rgb(255,0,0)"
* Text Alignment
* The text-align property is used to set the horizontal alignment of a text.
* A text can be left or right aligned, centered, or justified.
* text-align
* text-align-last
* direction
* unicode-bidi
* vertical-align
* Text Decoration
* The text-decoration-line property is used to add a decoration line to text.
* text-decoration-line
* text-decoration-color
* text-decoration-style
* text-decoration-thickness
* text-decoration
* Text Transformation
* The text-transform property is used to specify uppercase and lowercase letters in a text.
* It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word
* Text Spacing
* text-indent
* letter-spacing
* line-height
* word-spacing
* white-space

**CSS FONTS:**

* Choosing the right font for your website is important!
* Font Selection is Important
* Choosing the right font has a huge impact on how the readers experience a website.
* The right font can create a strong identity for your brand.
* Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.
* Generic Font Families
* Serif fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
* Sans-serif fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
* Monospace fonts - here all the letters have the same fixed width. They create a mechanical look.
* Cursive fonts imitate human handwriting.
* Fantasy fonts are decorative/playful fonts.
* Some Font Examples

|  |
| --- |
|  |

**CSS ICONS:**

* Icons can easily be added to your HTML page, by using an icon library.
* Adding Icons
* The simplest way to add an icon to your HTML page, is with an icon library, such as Font Awesome.
* Add the name of the specified icon class to any inline HTML element (like <i> or <span>).
* All the icons in the icon libraries below, are scalable vectors that can be customized with CSS (size, color, shadow, etc.)
* Font Awesome Icons
* To use the Font Awesome icons, go to fontawesome.com, sign in, and get a code to add in the <head> section of your HTML page:
* <script src="https://kit.fontawesome.com/yourcode.js" crossorigin="anonymous"></script>
* Example

|  |  |
| --- | --- |
| Example | Output |
| <html>  <meta name="viewport" content="width=device-width, initial-scale=1">  <script src="https://kit.fontawesome.com/a076d05399.js" crossorigin="anonymous"></script><body>  <p>Some Font Awesome icons:</p>  <i class="fas fa-cloud"></i>  <i class="fas fa-heart"></i>  <i class="fas fa-car"></i>  <i class="fas fa-file"></i>  <i class="fas fa-bars"></i>  <p>Styled Font Awesome icons (size and color):</p>  <i class="fas fa-cloud" style="font-size:24px;"></i>  <i class="fas fa-cloud" style="font-size:48px;color:red;"></i>  </body></html> |  |

* Bootstrap Icons
* To use the Bootstrap glyphicons, add the following line inside the <head> section of your HTML page:
* <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
* Example

|  |  |
| --- | --- |
| Example | Output |
| <html>  <meta name="viewport" content="width=device-width, initial-scale=1">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">  <body class="container">  <h3>Bootstrap icon library</h3>  <p>Some Bootstrap icons:</p>  <i class="glyphicon glyphicon-cloud"></i>  <i class="glyphicon glyphicon-remove"></i>  <i class="glyphicon glyphicon-user"></i>  <i class="glyphicon glyphicon-envelope"></i>  <i class="glyphicon glyphicon-thumbs-up"></i>  <br><br>  <p>Styled Bootstrap icons (size and color):</p>  <i class="glyphicon glyphicon-cloud" style="font-size:24px;"></i>  <i class="glyphicon glyphicon-cloud" style="font-size:36px;"></i>  <i class="glyphicon glyphicon-cloud" style="font-size:48px;color:red;"></i>  </body>  </html> |  |

* Google Icons
* <link rel="stylesheet"
* href="https://fonts.googleapis.com/icon?family=Material+Icons">

**CSS LINKS:**

* With CSS, links can be styled in many different ways.
* Example

|  |
| --- |
|  |

* Styling Links
* Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  a { color: hotpink; }  </style><body>  <p><b><a href="default.asp" target="\_blank">This is a link</a></b></p></body></html> |  |

* In addition, links can be styled differently depending on what state they are in. The four links states are:
* a:link - a normal, unvisited link
* a:visited - a link the user has visited
* a:hover - a link when the user mouses over it
* a:active - a link the moment it is clicked

**CSS LISTS:**

* The CSS list properties allow you to:
* Set different list item markers for ordered lists
* Set different list item markers for unordered lists
* Set an image as the list item marker
* Add background colors to lists and list items
* Different List Item Markers
* The list-style-type property specifies the type of list item marker.

|  |  |
| --- | --- |
| Example | Output |
| <html>  <style>  ul.a {  list-style-type: circle;}  ul.b {  list-style-type: square;}  ol.c {  list-style-type: upper-roman;}  ol.d {  list-style-type: lower-alpha;}  </style>  <body>  <p>Example of unordered lists:</p>  <ul class="a">  <li>Kamali</li>  <li>3sha</li>  </ul>  <ul class="b">  <li>Kamali</li>  <li>3sha</li></ul>  <p>Example of ordered lists:</p>  <ol class="c">  <li>kala</li>  <li>mala</li>  </ol>  <ol class="d">  <li>kalai</li>  <li>selvi</li>  </ol>  </body>  </html> |  |

**CSS TABLES:**

* Table Borders
* To specify table borders in CSS, use the border property.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  table, th, td {  border: 1px solid;}  </style><body><table>  <tr><th>Firstname</th>  <th>Lastname</th></tr>  <tr><td>Kamali</td>  <td>A</td>  </tr></table></body>  </html> |  |

* Full-Width Table
* The table above might seem small in some cases.
* If you need a table that should span the entire screen (full-width), add width: 20% to the <table> element.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  table, th, td {  border: 1px solid;}  table { width: 20%; }  </style><body>  <h2>Full-width Table</h2>  <table><tr>  <th>Firstname</th>  <th>Lastname</th>  </tr><tr>  <td>Peter</td>  <td>Griffin</td>  </tr></table>  </body></html> |  |

* Collapse Table Borders
* The border-collapse property sets whether the table borders should be collapsed into a single border

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  table, td, th {  border: 1px solid;}  table {width: 20%;  border-collapse: collapse;}  </style><body>  <h2>table borders collapse</h2>  <table><tr><th>Firstname</th>  <th>Lastname</th></tr>  <tr><td>Kamali</td><td>A</td></tr>  <tr><td>Kaaviyaa</td><td>A</td></tr>  </table>  </body>  </html> |  |

**CSS NAVIGATION BAR:**

* Having easy-to-use navigation is important for any web site.
* With CSS you can transform boring HTML menus into good-looking navigation bars.
* Navigation Bar = List of Links
* A navigation bar needs standard HTML as a base.
* In our examples we will build the navigation bar from a standard HTML list.
* A navigation bar is basically a list of links, so using the <ul> and <li> elements makes perfect sense.

|  |  |
| --- | --- |
| Example | Output |
| <html><body><ul>  <li><a href="#home">Home</a></li>  <li><a href="#news">News</a></li>  <li><a href="#contact">Contact</a></li>  <li><a href="#about">About</a></li>  </ul></body></html> |  |

* Now let's remove the bullets and the margins and padding from the list.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  ul {list-style-type: none;  margin: 0;  padding: 0;}  </style><body><ul>  <li><a href="#home">Home</a></li>  <li><a href="#news">News</a></li>  <li><a href="#contact">Contact</a></li>  <li><a href="#about">About</a></li>  </ul></body></html> |  |

**CSS DROPDOWNS:**

* Create a hoverable dropdown with CSS.
* Create a dropdown box that appears when the user moves the mouse over an element.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  .dropdown { position: relative;  display: inline-block;}  .dropdown-content { display: none;  position: absolute;  background-color: #f9f9f9;  min-width: 160px;  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);  padding: 12px 16px;  z-index: 1;}  .dropdown:hover .dropdown-content {  display: block;}  </style><body>  <h2>Hoverable Dropdown</h2>  <div class="dropdown">  <span>Mouse over me</span>  <div class="dropdown-content">  <p>Hii..This is Kamali</p></div>  </div></body></html> |  |

* Example Explained
* HTML:Use any element to open the dropdown content, e.g. a <span>, or a <button> element.
* Use a container element (like <div>) to create the dropdown content and add whatever you want inside of it.
* Wrap a <div> element around the elements to position the dropdown content correctly with CSS.
* CSS:The .dropdown class uses position:relative, which is needed when we want the dropdown content to be placed right below the dropdown button (using position:absolute).
* The .dropdown-content class holds the actual dropdown content. It is hidden by default, and will be displayed on hover (see below). Note the min-width is set to 160px. Feel free to change this. Tip: If you want the width of the dropdown content to be as wide as the dropdown button, set the width to 100% (and overflow:auto to enable scroll on small screens).
* Instead of using a border, we have used the CSS box-shadow property to make the dropdown menu look like a "card".

**CSS IMAGE GALLERY:**

* CSS can be used to create an image gallery.

|  |  |
| --- | --- |
| Example | Output |
| <html>  <style>  div.gallery {  margin: 5px;  border: 1px solid #ccc;  float: left;  width: 180px;}  div.gallery:hover {  border: 1px solid #777;}  div.gallery img {  width: 100%;  height: auto;  }  div.desc {  padding: 15px;  text-align: center;}  </style>  <body>  <div class="gallery">  <a target="\_blank" href="img\_5terre.jpg">  <img src="rose.jpeg" alt="Cinque Terre" width="600" height="400">  </a>  <div class="desc">Add a description of the image here</div>  </div>  </body>  </html> |  |

**CSS IMAGE SPRITES:**

* An image sprite is a collection of images put into a single image.
* A web page with many images can take a long time to load and generates multiple server requests.
* Using image sprites will reduce the number of server requests and save bandwidth.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  #home {  width: 46px;  height: 44px;  background: url(img\_navsprites.gif) 0 0;}  #next {  width: 43px;  height: 44px;  background: url(img\_navsprites.gif) -91px 0;}  </style><body>  <img id="home" src="img\_trans.gif"width="1" height="1">  <img id="next" src="img\_trans.gif" width="1" height="1">  </body></html> |  |

* Example explained:
* <img id="home" src="img\_trans.gif"> - Only defines a small transparent image because the src attribute cannot be empty. The displayed image will be the background image we specify in CSS
* width: 46px; height: 44px; - Defines the portion of the image we want to use
* background: url(img\_navsprites.gif) 0 0; - Defines the background image and its position (left 0px, top 0px)

**CSS ATTRIBUTE SELECTORS:**

* Style HTML Elements With Specific Attributes
* It is possible to style HTML elements that have specific attributes or attribute values.
* CSS Attribute Selectors

|  |
| --- |
|  |

* CSS [attribute] Selector
* The [attribute] selector is used to select elements with a specified attribute.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  a[target] {  background-color: yellow; }  </style><body>  <h2>CSS [attribute] Selector</h2>  < a href="https://www.w3schools.com">  w3schools.com</a>  <a href="http://www.disney.com" target="\_blank">disney.com</a>  <a href="http://www.wikipedia.org" target="\_top">wikipedia.org</a>  </body></html> |  |

* CSS [attribute="value"] Selector
* The [attribute="value"] selector is used to select elements with a specified attribute and value.

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  a[target="\_blank"] {  background-color: yellow;}  </style><body>  <h2>CSS [attribute="value"] Selector</h2>  <a href="https://www.w3schools.com">w3schools.com</a>  <a href="http://www.disney.com" target="\_blank">disney.com</a>  <a href="http://www.wikipedia.org" target="\_top">wikipedia.org</a>  </body>  </html> |  |

* CSS [attribute ~ = "value"] Selector
* The [attribute~="value"] selector is used to select elements with an attribute value containing a specified word.

|  |  |
| --- | --- |
| Example | Output |
| <html>  <style>  [title~="flower"] {  border: 5px solid yellow;}  </style>  <body>  <h2>CSS [attribute~="value"] Selector</h2>  <img src="klematis.jpg" title="klematis flower" width="150" height="113">  <img src="img\_flwr.gif" title="flower" width="150" height="200">  <img src="img\_tree.gif" title="tree" width="150" height="300">  </body>  </html> |  |

**CSS FORMS:**

* Using CSS to style an HTML Form

|  |  |
| --- | --- |
| Example | Output |
| <html>  <style>  input[type=text], select {  width: 100%;  padding: 12px 20px;  margin: 8px 0;  display: inline-block;  border: 1px solid #ccc;  border-radius: 4px;  box-sizing: border-box;}  input[type=submit] {  width: 100%;  background-color: #4CAF50;  color: white;  padding: 14px 20px;  margin: 8px 0;  border: none;  border-radius: 4px;  cursor: pointer;}  input[type=submit]:hover {  background-color: #45a049;}  div {  border-radius: 5px;  background-color: #f2f2f2;  padding: 20px;}  </style>  <body>  <div>  <form action="/action\_page.php">  <label for="fname">First Name</label>  <input type="text" id="fname" name="firstname" placeholder="Your name..">  <br>  <label for="lname">Last Name</label>  <input type="text" id="lname" name="lastname" placeholder="Your last name..">  <br>  <label for="country">Country</label>  <select id="country" name="country">  <option value="australia">Australia</option>  <option value="canada">Canada</option>  <option value="usa">USA</option>  </select>  <input type="submit" value="Submit">  </form>  </div></body></html> |  |

**CSS COUNTERS:**

* CSS counters are "variables" maintained by CSS whose values can be incremented by CSS rules (to track how many times they are used).
* Counters let you adjust the appearance of content based on its placement in the document.
* Automatic Numbering With Counters
* CSS counters are like "variables". The variable values can be incremented by CSS rules (which will track how many times they are used).
* To work with CSS counters we will use the following properties:
* counter-reset - Creates or resets a counter
* counter-increment - Increments a counter value
* content - Inserts generated content
* counter() or counters() function - Adds the value of a counter to an element
* To use a CSS counter, it must first be created with counter-reset.
* The following example creates a counter for the page (in the body selector), then increments the counter value for each <h2> element and adds "Section <value of the counter>:" to the beginning of each <h2> element:

|  |  |
| --- | --- |
| Example | Output |
| <html><style>  body {  counter-reset: section;}  h2::before {  counter-increment: section;  content: "Section " counter(section) ": ";}  </style><body><u>  <h1>Using CSS Counters</h1></u>  <h2>Mala</h2>  <h2>Kala</h2>  <h2>Kalai</h2>  <h2>Suga</h2>  <h2>Sun</h2>  </body></html> |  |

* Nesting Counters

The following example creates one counter for the page (section) and one counter for each <h3> element (subsection). The "section" counter will be counted for each <h3> element with "Section <value of the section counter>.", and the "subsection" counter will be counted for each <h4> element with "<value of the section counter>.<value of the subsection counter>".

|  |  |
| --- | --- |
| Example | Output |
| <html>  <style>  body {  counter-reset: section;}  h3::before {  counter-increment: section;  content: "Section " counter(section) ". ";}  h4::before {  counter-increment: subsection;  content: counter(section) "." counter(subsection) " ";}  </style><body>  <h3>flowers</h3><h4>Rose</h4><h4>Lotus</h4>  <h3>fruits</h3><h4>Mango</h4><h4>Apple</h4>  </body>  </html> |  |

* A counter can also be useful to make outlined lists because a new instance of a counter is automatically created in child elements. Here we use the counters() function to insert a string between different levels of nested counters:

|  |  |
| --- | --- |
| Example | Output |
| <html>  <style>  ol {  counter-reset: section;  list-style-type: none;}  li::before {  counter-increment: section;  content: counters(section,".") " "; }  </style>  <body>  <ol>  <li>FLOWER</li>  <li>SNACKS  <ol>  <li>Samosa</li>  <li>Panipuri</li>  <li>Fruit  <ol>  <li>Apple</li>  <li>Orange</li>  <li>Banana</li>  </ol>  </li>  <li>Vada pav</li>  </ol>  </li>  <br>  <li>VEGETABLES</li>  </ol>  </body>  </html> |  |

* CSS Counter Properties

|  |
| --- |
|  |