

What is CSS?

- CSS stands for Cascading Style Sheets
 - 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
-

InlineCss

```
<h1 style="color:blue;">HelloWorld</h1>  
<p style="color:tomato;">Loremipsum...</p>
```

InternalCss

```
<Style>  
    h1 {  
        color:Blue;  
    }  
    p {  
        color:Tomato;  
    }  
</Style>  
<h1>HelloWorld</h1>  
<p>Loremipsum...</p>
```

ExternalCss

```
<!Doctypehtml>

<html>

<head>

    <title>ExternalCss</title>

    <linkhref="css/text.css" rel="stylesheet">

</head>

<body>

    <h1>HelloWorld</h1>

    <p>Loremipsum...</p>

</body>

</html>h
1 {
    color:blue;
}
p {
    color:tomato;
}
```

Color

```
<h1style="color:blue;">HelloWorld</h1>

<pstyle="color:tomato;">Loremipsum...</p>
```

type:

RGB,HEX,HSL.

RGB:

```
<h1 style="color:rgb(255,165,0);">rgb(255,165, 0)</h1>
```

HEX:

```
<h1 style="color:#ee82ee;">rgb(255,0,0)</h1>
```

HSL:

```
<h1 style="color:hsl(300,76%,72%);">hsl(300, 76%,72%)</h1>
```

Background

```
<Style>
```

```
h1 {
```

```
background-image: url("img_flwr.gif");
```

```
background-position:rightbottom,lefttop;
```

```
background-repeat: no-repeat, repeat;
```

```
background-size: 100px 80px;
```

```
background-color: Tomato;
```

```
background-origin: border-box;
```

```
background-clip: content;
```

```
background-blend-mode:darken;
```

```
background-attachment: fixed;
```

```
</Style>
```

<h1>Loremipsum...</h1>

Property:

Background-Color White	Background-clip Border-box,content-box,padding-box.		Background-image None		Background-size Auto,contain,Cover
Background-blend-mode Color,color-burn,color-image,darken,difference,exclusion, hard-light, hue, lighten,luminosity, multiply,normal,overlay,saturation,screen,soft,light	Background-position Bottom,center,left,right, top		Background-repeat No-repeat,repeat,round,space		Background-origin Border-box,content-box,padding-box
	Background-position-x Center,left, right	Background-position-y Bottom,top, center	Background-repeat -x No-repeat, repeat	Background-repeat -y No-repeat, repeat	background-attachment fixed,local,scroll, unset

Border

<Style>

```
h1 {  
  
    border:2pxsolidblack;  
    border-radius: 5px;  
}  
h2{  
  
    border-top: 2pxsolidred;  
    border-bottom:2pxdottedblue;  
}  
p{  
  
    border-style:dotted;  
    border-width: 2px;  
    border-color: black;  
}
```

```

.text{
    border-image:url("border.png")30round;
    border-collapse: separate;
    border-spacing: 15px 50px;
    border-top-right-radius: 25px;
    border-right-bottom-radius:25px;
}

.text{
    border:10pxsolidtransparent; padding:
    15px;
    border-image:url(.,image.png')30round;
}

</Style>

<divclass="text">

    <h1>Hello World</h1>

    <h2>Heading2</h2>

    <p>text </p>

</div>

```

Property:

| Border-style
Solid,Dashed,Dotted,
Double,Groove,
Hidden,Inset,None,
Outset,Ridge. | Border-width
Medium,Thick,Thin. | Border-Image
None,Repeat,Round,Space,Starch. | | | | | Border-Color |
|---|------------------------------------|---|--------------------------|---------------------------|------------------------------------|--------------------------|--------------|
| | | Border
Image
Source | Border
Image
Slice | Border
Image
Outset | Border
Image
Repeat | Border
Image
Width | |
| | | None | Number
px,%.
 | Number
px,%.
 | Repeat
round
space
starch | Auto | |

| | | | | |
|---|--|---|--|--|
| Border-radius

revert,unset,initial,inherit | Border-inline-start

None,revert,unset,initial,inherit | Border-inline-end

None,revert,unset,initial,inherit | Border-block-Start

None,revert,unset,initial,inherit | Border-block-end

None,revert,unset,initial,inherit |
| Width

Medium,thin,thin | Style

Dashed,dotted,double,groove,hidden,insert,none,outset,ridge,solid | | Color | |
| Border-left

Dashed,dotted,double,groove,hidden,insert,medium,none,outset,ridge,solid,thick,thin | Border-right

Dashed,dotted,double,groove,hidden,insert,medium,none,outset,ridge,solid,thick,thin | Border-bottom

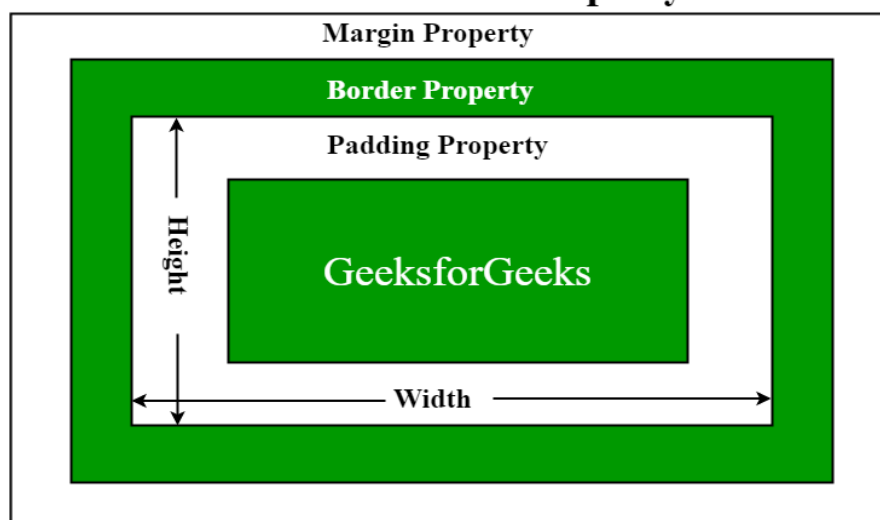
Dashed,dotted,double,groove,hidden,insert,medium,none,outset,ridge,solid,thick,thin | Border-top

Dashed,dotted,double,groove,hidden,insert,medium,none,outset,ridge,solid,thick,thin | |
| Width

Medium,thin,thin | Style

Dashed,dotted,double,groove,hidden,insert,none,outset,ridge,solid | | Color | |

CSS Box-Model Property



Margin

```
<Style>
  h1 {
    margin:10px;
    margin-top:25px;
    margin:left:10px;
    margin:25px10px;
    background-color:yellow;
  }
  .text{
    background-color:green;
  }
</Style>
<divclass="text">
  <h1>Hello World</h1>
  <h2>Heading2</h2>
  <p>text </p>
</div>
```

Property:

margin-top,right,bottom, left.

Padding

```
<Style>
  h1 {
    padding:10px;
    padding-top:25px;
    padding:left:10px;
    padding:25px10px;
    background-color:yellow;
  }
```

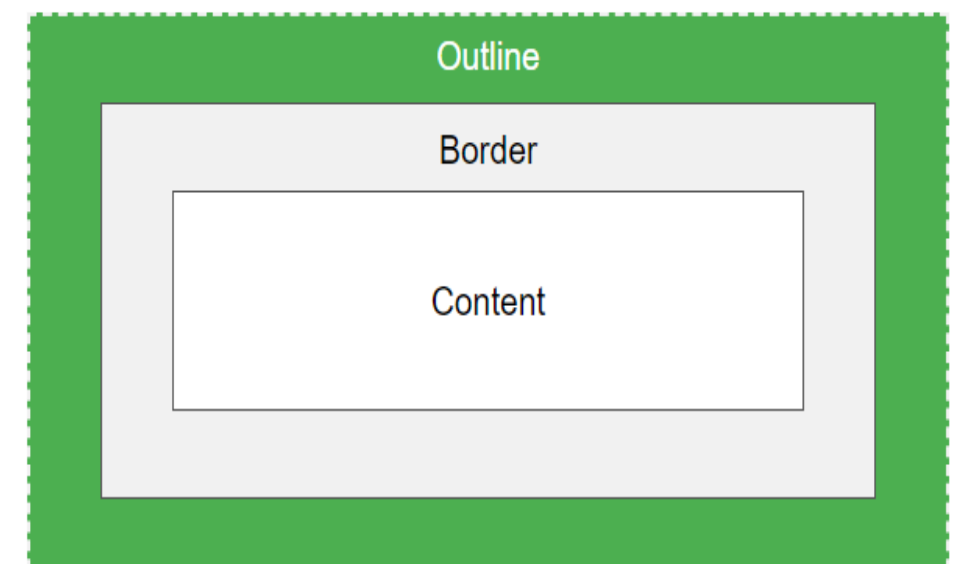
```

        .text{
            background-color:green;
        }
</Style>
<divclass="text">
    <h1>Hello World</h1>
    <h2>Heading2</h2>
    <p>text </p>
</div>

```

Property:

padding– top,right, bottom, left.



Outline

```

<Style>
    .text{
        border: 2px solid black;
        outline: 10px solid #ff0;
    }
</Style>

```



```

<divclass="text">
    <h1>Hello World</h1>
</div>

```

Property:

Outline			
Outline-style	Outline-color	Outline-width	Outline-offset

Text

```

<Style>
    .h1 {
        text-align:center;
        text-shadow:2px2pxyellow;
        text-align:uppercase;
    }
    .text{
        width: 300px;
        height:100px;
        background-color:green;
        box-shadow:10px10px5px2pxgrey;
    }
</Style>
<divclass="text">
    <h1>Hello World</h1>
    <p></p>
</div>

```

Property:

Color	Direction	Text-align	Decoration	White-space
	rtl,ltr	top,left,right,center,	blink,dashed,dotted,double,line-	break-spaces,none,

		justify,start,end,	through,none,overline,unset,wavy,solid		normal,no-wrap,pre,pre-line, pre-wrap
Transformation uppercase, lowercase,capitalize	Spacing letter-spacing, line-height,word-spacing	Shadow			Text-indent
		text		box	
		none,h-offset,v-offset,blur,spread,color, inset			

Definition:

text-indent: Specifies the indentation of the first line in a text-block.

Font

```

<Style>
    .h1 {
        font-family:"Sofia";
        font-size: 22px;
    }
    P {
        font:italic small-caps bold 12px/30px Georgia, serif;
    }
</Style>
<div class="text">
    <h1>Hello World</h1>
    <p>This paragraph</p>
</div>

```

Property:

font –family	font-style	font-size	font-weight	font-variant
sans-serif, serif, serif (red serifs,	normal, italic, oblique	(number)px, %, em, rem, vw	normal, 100, 200, 300, 400, 500, 600, 700, 800, 900, bold, bolder, lighter	normal, small-caps

Lists

```
<Style>
    .ul{
        list-style:squareinsideurl("sqpurple.gif");
        list-type: upper-alpha;
    }
</Style>
<ul>
    <li>Coffee</li>
    <li>Tea</li>
    <li>CocaCola</li>
</ul>
```

Property:

List-style		
Square		
List-style-type circle,square,upper-roman,lower-alpha	List-style-image url, none	List-style-position outside,inside

Table

```
<Style>
    caption-side:bottom;
</Style>
<tableid="example1"border="1">
    <caption>Table1.1 Customers</caption>
    <thead>
        <tr>
            <th>Company</th>
            <th>Contact</th>
            <th>Country</th>
        </tr>
    </thead>
```

```

        <tbody>
            <tr>
                <td>Alfreds Futterkiste</td>
                <td>MariaAnders</td>
                <td>Germany</td>
            </tr>
        </tbody>
    </table>

```

Property:

Top,bottom	Caption-side
------------	--------------

Display

```

<Style>
    ulli{
        display:flex;
        display:none;
    }
</Style>
<ul>
    <li>Coffee</li>
    <li>Tea</li>
    <li>CocaCola</li>
</ul>

```

Property:

1.Display:flex

The flex property sets the flexible length on flexible items.

. flex-direction: row, row-reverse, column, column-reverse, initial, inherit;

. flex-wrap: nowrap, wrap, wrap-reverse ,initial , inherit;

. flex-flow: row-reverse wrap;

Positional alignment

justify-content: center, left, right, space-between, space-around, space-evenly, unset;

align-items: center,baseline,start,end,unset;

gap: 20px;

order:

```
<div>
  <div style="order: 2">Item 1</div>
  <div style="order: 3">Item 2</div>
  <div style="order: 1">Item 3</div>
</div>
```

2.Display:grid

A grid can be defined as an intersecting set of vertical and horizontal lines. CSS Grid layout separates a page into major sections.

grid-template-columns: none, auto, repeat(3,1fr), also give in percentage(%);

grid-template-rows: none, auto, repeat(3,1fr), also give in percentage(%);

grid-column: 1 / span 2;

Positional alignment

justify-content: center, left, right, space-between, space-around, space-evenly, unset;

align-items: center,baseline,start,end,unset;

gap: 20px;

grid-column-gap: 50px;

grid-row-gap: 10px;

order:

```
<div>
  <div style="order: 2">Item 1</div>
  <div style="order: 3">Item 2</div>
  <div style="order: 1">Item 3</div>
</div>
```

3.Display: block

Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width

4.Display: inline

Displays an element as an inline element (like). Any height and width properties will have no effect. This is default.

5.Display: inline-block

Displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values.

6.Display: none

The element is completely removed.

Visibility

```
<Style>
  ul li{
    visibility:visible;
  }
</Style>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>CocaCola</li>
</ul>
```

Property:

visible,hidden,collapse	visiblity
-------------------------	-----------

Z-index

```
<Style>
    .text{
        z-index:1;
    }
</Style>
<divclass="text">
    <h1>Hello World</h1>
    <p></p>
</div>
```

Overflow

```
<Style>
    table{
        overflow:auto;
    }
```

```

</Style>
<tableid="example1"border="1"style="min-width:750px;">
  <caption>Table1.1 Customers</caption>
  <thead>
    <tr>
      <th>Company</th>
      <th>Contact</th>
      <th>Country</th>
    </tr>
    <tr>
      <th>Company</th>
      <th>Contact</th>
      <th>Country</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Alfreds Futterkiste</td>
      <td>MariaAnders</td>
      <td>Germany</td>
    </tr>
    <tr>
      <td>Alfreds Futterkiste</td>
      <td>MariaAnders</td>
      <td>Germany</td>
    </tr>
    <tr>
      <td>Alfreds Futterkiste</td>
      <td>MariaAnders</td>
      <td>MariaAnders</td>
    </tr>
  </tbody>
</table>

```


Property:

overflow			
visible	hidden	scroll scroll-x,scroll-y	auto

Float

```
<Style>
  h1 {
    float:left;

  }
  p{
    float:right;
  }
</Style>
<divclass="text">
  <h1>Hello World</h1>
  <p></p>
</div>
```

Property:

Float
Left,right,none

Clear

```
<Style>
  h1 {
    float:left;
  }
  p{
    clear:left;
  }
</Style>
```

```

<divclass="text">
    <h1>Hello World</h1>
    <p></p>
</div>

```

Property:

clear
Left,right,none,both

Opacity

```

<Style>
    img{
        opacity: 0.5;
    }
</Style>
<divclass="text">
    <imgsrc="img_forest.jpg"alt="Forest"width="170px"height="100px">
</div>

```

Position

```

<Style>
    div{
    }
    h2 { position:relative;

        position:absolute;
        left:100px;
        top:150px;
    }
</Style>
<divclass="text">
    <h2>Thisisaheadingwithanabsolute position</h2>
</div>

```

Property:

Position				
position-relative top,bottom,left, right	Position-absolute top,bottom,left, right	position-sticky top,bottom,left, right	position-fixed top,bottom,left, right	position-static default

Object-fit

```
<Style>
    .fill{
        object-fit:fill;
    }
    .contain{
        object-fit:contain;
    }
    .cover{
        object-fit:cover;
    }
    .scale-down{
        object-fit:scale-down;
    }
    .none{
        object-fit:none;
    }
</Style>
<divclass="text">
    <h1>Theobject-fitProperty</h1>
    <imgsrc="paris.jpg"alt="Paris"style="width:200px;height:400px;">
    <imgclass="fill"src="paris.jpg"alt="Paris"style="width:200px; height:400px;">
```

```

<imgclass="contain"src="paris.jpg"alt="Paris"style="width:200px;height:400px;">
<imgclass="cover"src="paris.jpg"alt="Paris"style="width:200px; height:400px;">
<imgclass="scale-down"src="paris.jpg"alt="Paris"style="width:200px; height:400px;">
<imgclass="none"src="paris.jpg"alt="Paris"style="width:200px;height:400px;">

```

</div>

Property:

fill,contain,cover,none,scale-down	Object-fit
------------------------------------	------------

MultipleColumns

<Style>

```

.newspaper{
    column-count:3;
    column-gap: 40px;
    column-rule-style:solid;
    column-rule-width:1px;
    column-rule-color:lightblue;
    column-width: 100px;
}

```

```

h2 {
    column-span: all;
}

```

```

.newspaper2{
    column-fill: balance;
}

```

</Style>

<divclass="newspaper">

<h2>Lorem IpsumDolorSitAmet</h2>

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis

nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

</div>

<h2>column-fill:balance(default):</h2>

<divclass="newspaper2">

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat...

</div>

Property:

column

column-count, column-gap, column-rule-style, column-rule-width, column-rule-color, column-rule, column-span, column-width, column-fill, column-rule, columns

Input

<style>

```
input[type=text]{
    border:2pxsolidred;
    border-radius: 4px;
}
input[type=text]:focus{
    background-color:lightblue;
}
```

</style>

<form>

<labelfor="fname">FirstName</label>

```

    <input type="text" id="fname" name="fname" value="John">
    <label for="lname">Last Name</label>
    <input type="text" id="lname" name="lname" value="Doe">
</form>

```

Property:

input[type=text]	input[type=password]	input[type=number]
------------------	----------------------	--------------------

Definition:

input[type=text] - will only select text fields.

input[type=password] - will only select password fields.

input[type=number] - will only select number fields.

Combinators

Basic:

```

<style>
    divp {
        background-color: yellow;
    }
    div>p {
        background-color: green;
    }
    div~p {
        background-color: red;
    }
    div+p {
        background-color: blue;
    }
</style>

```

<h2>Descendant Selector</h2>

<p>The descendant selector matches all elements that are descendants of a specified element.</p>

<div>

<p>Paragraph 1 in the div.</p>

```

        <p>Paragraph2inthe div.</p>
</div>
<section>
        <p>Paragraph3inthe div.</p>
</section>
<p>Paragraph4.Notina div.</p>
<p>Paragraph5.Notina div.</p>

```

Type:

1. descendantselector(space)
 2. childselector(>)
 3. adjacentsiblingsselector(+)
 4. generalsiblingsselector(~)
-

Selector

```

<style>
    text1,text2 {Background-color:yellow;}
    .text,p{
        background-color:red;
    }
    h2:hover{
        background-color:blue;
    }
    p:active{
        background-color:orange;
    }
    p:nth-child(even){
        color:red;
    }
    p:nth-child(odd){
        Color:green;
    }
    p:nth-child(n+1){
        color:blue;
    }
    p:nth-child(2n+1){
        color:pink;
    }

```

```

</style>
<h2>DescendantSelector</h2>
<divclass="text1">
  <pclass="text1">Paragraph1inthediv.</p>
  <pclass="text2">Paragraph2inthe div.</p>
  <p>Paragraph3inthediv</p>
  <p>paragraph4inthediv</p>
  <p>paragraph5inthediv</p>
</div>
<p>Paragraph3inthe div.</p>

```

Type:

class	class1.class2	class1class2	#id	*	element1	element2, class1	element1,element2	element1element2	element1>element2	
element1+element	element1~element2	[attribute]	[attribute~value]	[attribute value]	[attribute^=value]	[attribute\$=value]	[attribute*=value]	:active	::after	
::before	:checked	:default	:default	:disabled	:empty	:enabled	:first-child	::first-child	::first-letter	
::first-letter	::first-line	:first-of-type	:focus	:full screen	:hover	:in-range	:indeterminate	:invalid	:lang(language)	
:last-child	:last-of-type	:link	:not(selector)	:nth-child(n)	:nth-last-child(n)	:nth-last-of-type(n)	:nth-of-type(n)	:only-of-type	:only-child	
:optional	:out-of-range	::placeholder	:read-only	:read-write	:required	:root	::selection	:target	:valid	:visited

AttributeSelectors

```

<style>
  a[target]{
    background-color:yellow;
  }
  [class^="test"]{
    background:#ffff00;
  }
  [lang|=en]{
    background-color:red;
  }
</style>

```

```
<h2>DescendantSelector</h2>
```



```

<divclass="text1">
    <a href="http:// https://sohamsolution.com/"lang="en" class="text_en">soham.com</a><a
    href="http://www.facebok.com"target="_blank"class="text"lang="hindi">facebook.com</a>
    <ahref="http://www.wikipedia.org"target="_top" class="text2">wikipedia.org</a>
</div>

```

Type:

[attribute],[attribute=value],[attribute~=value],[attribute|=value],[attribute^=value], [attribute\$=value],
 [attribute*=value]

PseudoClasses

```

<style>
    divp:first-child {
        background-color:yellow;
    }
    divp:last-child {
        background-color:yellow;
        color:white;
    }
    sectionp{
        background-color:red;
    }
    sectionp:hover{
        background-color:blue;
    }
    p:active{
        background-color:orange;
    }

    :has() selector for border styling

    ul li:has(input:checked) {

        border: 2px solid #f806e4;
    }

```

is() is used for selecting multiple elements in more compact form.

```
.contain :is(h1, h2, h3,) {  
    color: red;  
}
```

:focus - :focus: to bring attention to the element that currently has the focus.

```
.element:focus-visible {  
  
    background-color: pink;  
  
}
```

</style>

<h2>DescendantSelector</h2>

<p>Thedescendantsselector matchesall elementsthataredescendantsof aspecifiedelement.</p>

```
<div>
  <p>Paragraph1inthe div.</p>
  <p>Paragraph2inthe div.</p>
</div>
<section>
  <p>Paragraph3inthe div.</p>
</section>
```

Type:

active,:checked,:disabled,:empty,:enabled,:first-child,:first-of-type,:focus,:hover,:in-range,
:invalid,:lang(language),:last-child,:link,:not(selector),:nth-child(n),:nth-last-child(n),:nth-of-
type(n),:only-of-type,:only-child,:optional,:out-of-range,:read-only,:read-write,:required,:root,
:target,:valid, :visited

PseudoElements

```
<style>
  p::first-line{
    color:#ff0000;
    font-variant:small-caps;
  }
  p::first-letter{
    color: #ff0000;
    font-size:xx-large;
  }
  h1::before{
    content:url(smiley.gif);
  }
  ::selection{
    color: red;
    background:yellow;
```

```

    }
</style>

<h2>DescendantSelector</h2>

<p>Thedescendantselectormatchesallelementsthataredescendantsofaspecified element.</p>

<div>

    <p>Paragraph1inthe div.</p>
    <p>Paragraph2inthe div.</p>
    <p>Paragraph3inthe div.</p>

</div>

<p>Paragraph4.Notina div.</p>
<p>Paragraph5.Notina div.</p>

```

Type:

::after, ::before, ::first-letter, ::first-line, ::selection

Dropdown

```

<style>

.dropdown{
    position: relative;
    display:inline-block;
}

.dropdown-content{
    display: none;
    position:absolute;
    background-color:#f9f9f9;
    min-width: 160px;
    box-shadow:0px8px16px0pxrgba(0,0,0,0.2); padding:
    12px 16px;
    z-index:1;
}

.dropdown:hover.dropdown-content{

```

```

        display:block;
    }
</style>
<divclass="dropdown">
    <span>Mouseoverme</span>
    <divclass="dropdown-content">
        <p>HelloWorld!</p>
    </div>
</div>

```

Tooltip

```

<style>
.tooltip {
    position:relative;display:inline-block;border-bottom:1px dottedblack;
}
.tooltip.tooltiptext{
    visibility:hidden;
    width: 120px;
    background-color:black;
    color: #fff;
    text-align: center;
    border-radius:6px;
    padding: 5px 0;
    position: absolute;
    z-index: 1;
    top: 25px;
    right:-15px;
}
.tooltiptext::after{
    content: "";
    position:absolute;
    bottom: 100%;
    left:50%;

```

```

        margin-left: -5px;
        border-width:5px;
        border-style:solid;
        border-color:transparenttransparent#555 transparent;
    }
    .tooltip:hover.tooltiptext{
        visibility:visible;
    }
</style>
<divclass="tooltip">Hoveroverme
    <spanclass="tooltiptext">Tooltip text</span>
</div>

```

Gradients

```

<style>
    #grad1 {
        height: 55px;
        background-color:red;
        background-image:linear-gradient(to right,red,orange,yellow,green,blue,indigo, violet);
    }
    h1 {
        text-align:center;
        margin:auto;
        color:#888888;
        font-size:40px;
        font-weight:bold;
        background:repeating-radial-gradient(red, yellow10%,green15%);
    }
    #grad2 {
        background:radial-gradient(farthest-side at 60% 55%,red,yellow, black);
    }
</style>

<divid="grad1">

```

```

        <h1 class>Rainbow Background</h1>
    </div>
    <div id="grad2">
        <h1 class>Rainbow Background</h1>
    </div>

```

Property:

linear gradients	repeating linear-gradient	radial gradients closest-side, farthest-side, closest- corner, farthest-corner	repeating-radial gradient
---------------------	------------------------------	---	------------------------------

Transforms

Basic:

```

<style>
    #div1 {
        height:150px;width:150px;margin-left:60px; border:
        1px solid black;
        perspective: 100px;
        perspective-origin:left;
    }
    #div2 {
        padding:50px;
        background:rgba(100,100,100,0.5);
        transform: rotateX(45deg);
    }
</style>

```

<h1>The rotateZ() Method</h1>

<p>The rotateZ() method rotates an element around its Z-axis at a given degree.</p>

<div id="div1">DIV1

<div id="div2">DIV2</div>

</div>

<div id="myDiv">

This div element is rotated 90 degrees.

</div>

Property and Method:

Transform	
transform, transform-origin, transform-style, perspective, perspective-origin, backface-visibility	
2d Transform	3d Transform
translate(), rotate(), rotateX(), rotateY(), scaleX(), scaleY(), scale(), skewX(), skewY(), skew(), matrix()	translate(), rotate(), rotateX(), rotateY(), rotateZ(), scaleX(), scaleY(), scaleZ(), scale(), skewX(), skewY(), skewZ(), skew(), matrix()

Transition

```
<style>
  div {
    width: 100px;
    height: 100px;
    background:red;
    transition: width 2s,height 4s;
    transition-timing-function:ease;
  }
  div:hover{
    width: 300px;
    height:300px;
  }
</style>
<div>
  <h1 style="color:white;">The transition Property</h1>
</div>
```

Property and Method:

Transition			
transition-delay	transition-duration	transition-property	transition-timing-function

			ease,linear,ease-in,ease-out,ease,in- out, cubic-bezier (n,n,n)
--	--	--	--

Definition:

ease-specifies a transition effect with a slow start, then fast, then end slowly (this is default)

linear-specifies a transition effect with the same speed from start to end

ease-in-specifies a transition effect with a slow start

ease-out-specifies a transition effect with a slow end

ease-in-out-specifies a transition effect with a slow start and end

cubic-bezier(n,n,n,n)-lets you define your own values in a cubic-bezier function

Animation

```
<style>
  div {
    width: 100px;
    height: 100px;
    background-color: red;
    position: relative;
    animation-name: example;
    animation-duration: 4s;
  }
  @keyframes example {
    0% {background-color: red; left: 0px; top: 0px; }
    50% {background-color: blue; left: 200px; top: 200px; } 100% {
      background-color: green; left: 0px; top: 0px; }
  }
</style>
<div>
  <h1 style="color: white; font-size: 14px;">The transition Property</h1>
</div>
```

Property and Method:

Animation

@keyframes, animation-name, animation-duration, delay, animation-iteration-count, animation-direction, animation-timing-function, animation-fill-mode, animationanimation-d

Definition:

ease-specifies a transition effect with a slow start, then fast, then end slowly (this is default)

linear-specifies a transition effect with the same speed from start to end

ease-in-specifies a transition effect with a slow start

ease-out-specifies a transition effect with a slow end

ease-in-out-specifies a transition effect with a slow start and end

cubic-bezier(n,n,n,n)-lets you define your own values in a cubic-bezier function

Variables-The var() Function

```
<style>
  :root {
    --blue: #1e90ff;
    --white: #fff;
  }
  body {
    background-color: var(--blue);
  }
  h2 {
    border-bottom: 2px solid var(--blue);
  }
  .container {
    color: var(--blue);
    background-color: var(--white);
    padding: 15px;
  }
  button {
    background-color: var(--white);
```

```

        color:var(--blue);
        border:1pxsolidvar(--blue); padding:
        5px;
    }
</style>
<h1>Usingthevar()Function</h1>
    <divclass="container">
        <h2>LoremIpsum</h2>
        <p>Loremipsumdolorsitamet,consecteturadipiscingelit.Etiamsemperdiamat erat
        pulvinar, at pulvinar felis blandit.</p>
        <p><button>Yes</button><button>No</button></p>
    </div>

```

Responsive- MediaQueries

```

<style>
    body{ background-color:lightgreen;

    }
    @mediaonlyscreenand(max-width:1200px){ body {
        background-color:lightblue;
    }
}
</style>

<!DOCTYPEhtml>
<html>
    <head>
        <metaname="viewport"content="width=device-width, initial-scale=1.0">
    </head>
    <body>

```

```

    <p>Resizethebrowserwindow.Whenthewidthofthisdocumentis600pixelsor less, the
    background-color is "lightblue", otherwise it is "lightgreen".</p>
  </body>
</html>

```

@media (max-width:1200px) (containerwidth:960px;)	@media (max-width: 991px) (containerwidth:750px;)	@media (max-width:767px) (containerwidth:550px;)	@media (max-width:575px) (containerwidth:100%; padding:0 15px;)
@media(max-width:375px) (containerwidth:100%;padding:015px;)		@media(max-width: 320px) (containerwidth:100%;padding:015px;)	

MediaTypes:

All,print,screen,speech

MediaFeatures:

any-hover,any-pointer,aspect-ratio,color,color-gamut,color-index,grid,height,
 hover, inverted-colors, light-level, max-aspect-ratio, max-color, max-color-index, max-height, max-
 monochrome, max-resolution, max-width, min-aspect-ratio, min-color, min-color-index, min-height, min-
 monochrome, min-resolution, min-width, monochrome, orientation, overflow-block, overflow-inline,
 pointer, resolution, scan, scripting, update, width

Definition:

All-Default.Usedforallmediatype devices

Print- Usedforprinters

screen-Usedforcomputerscreens,tablets,smart-phonesetc.

speech-Used forscreen readers that "reads"the pageout loud

webkit-scrollbar

```

<style>
  body {
    font-family: Tahoma, Arial, sans-serif;
    font-size: 15px;

```

```

}
p { margin-top: 0; }
html {
  overflow: auto;
}

body {
  position: absolute;
  top: 10px;
  left: 20px;
  bottom: 10px;
  right: 0;
  padding: 30px;
  overflow-y: scroll;
  overflow-x: hidden;
}

div { height: 1000px; }

/* Scrollbar Styling */
::-webkit-scrollbar {
  width: 10px;
}

::-webkit-scrollbar-track {
  background-color: #ebebeb;
  -webkit-border-radius: 10px;
  border-radius: 10px;
}

::-webkit-scrollbar-thumb {
  -webkit-border-radius: 10px;
  border-radius: 10px;
  background: #6d6d6d;
}

```

```

</style>
</head>
<body>
  <div>
    <p>Note this only works for webkit browsers (Chrome and Safari). Here's how you make your own custom scrollbar...</p>
    <ol>
      <li>Set html to overflow: auto</li>
      <li>Set body to overflow-y: scroll</li>
      <li>Create a div with a height larger than the window (just for testing)</li>
      <li>Set a width for ::-webkit-scrollbar</li>
      <li>Set a background-color for ::-webkit-scrollbar-track</li>
      <li>Set a background-color for ::-webkit-scrollbar-thumb</li>
    </ol>
    <p>Optional: Put a border-radius on the scrollbar and track for a nicer look</p>
  </div>
</body>

```

-webkit-mask-image

```

<style>
.mask1 {
  -webkit-mask-image: url(w3logo.png);
  mask-image: url(w3logo.png);
  -webkit-mask-repeat: no-repeat;
  mask-repeat: no-repeat;
}
</style>
</head>
<body>

<h1>The mask-image Property</h1>

<h3>An image with a mask layer image:</h3>
<div class="mask1">
  

```

```
</div>
```

```
<h3>Original image:</h3>
```

```

```

```
</body>
```

Text-overflow

```
<style>
```

```
div.a {  
  white-space: nowrap;  
  width: 50px;  
  overflow: hidden;  
  text-overflow: clip;  
  border: 1px solid #000000;  
}
```

```
div.b {  
  white-space: nowrap;  
  width: 50px;  
  overflow: hidden;  
  text-overflow: ellipsis;  
  border: 1px solid #000000;  
}
```

```
div.c {  
  white-space: nowrap;  
  width: 50px;  
  overflow: hidden;  
  text-overflow: "----";  
  border: 1px solid #000000;  
}
```

```
</style>
```

```
</head>
```

<body>

<h1>The text-overflow Property</h1>

<p>The following two divs contains a text that will not fit in the box.</p>

<h2>text-overflow: clip (default):</h2>

<div class="a">Hello world!</div>

<h2>text-overflow: ellipsis:</h2>

<div class="b">Hello world!</div>

<h2>text-overflow: "----" (user defined string):</h2>

<div class="c">Hello world!</div>

<p>Note: The text-overflow: "string" only works in Firefox.</p>

</body>
