



CASCADING STYLE SHEET

HANDWRITTEN NOTES

BY

@ code with - random

TOPICS

Date :
Page No. :

- 1 → Introduction
- 2 → Add CSS
- 3 → Selectors
- 4 → Comments
- 5 → Colors
- 6 → Background
- 7 → Borders
- 8 → Font
- 9 → Box Model
- 10 → Padding
- 11 → Margin
- 12 → Float
- 13 → Position
- 14 → Overflow
- 15 → ! Important
- 16 → Hover Effect
- 17 → Transform property
- 18 → Transition property
- 19 → Animation
- 20 → Responsive Design
- 21 → Viewport
- 22 → Responsive Viewport

→ Introduction

CSS stands for "Cascading Style Sheet". It is generally used with HTML to change the style of webpages and user interface. It allows you to change the size, color, font and style of the text; margin and padding; background and borders and so on. It can also be used to position element on a page.

It is used with any kind of XML document including XML, XUV and SVG.

→ Add CSS

Before writing the CSS, the first step is insert the CSS in HTML document. There are three methods for insert the CSS.

- i) Inline CSS
- ii) Internal CSS
- iii) External CSS

i) Inline CSS

Inline CSS is used for apply a style for single HTML element. It is written within any HTML element by using "Style" attribute.

Syntax

`<h1 style = " style property "> Content </h1>`

example

`<body>`

`h1 style = "Color: red" > inline CSS </h1>`

Element Inline CSS

ii) Internal CSS

Internal CSS is used to apply a style on HTML documents, In this you easily target multiple elements and class, id (which we will study in the section of selectors).

To write a CSS Internal, you place a `<style> ----- </style>` tag inside the HTML `<head> ----- </head>`.

example

```
<head>
  <style>
    h1 {
      color: red;
    }
  </style>
</head>

<body>
  <h1> Internal CSS </h1>
</body>
```

iii) External CSS

External CSS is separate CSS file that can be accessed by creating a link within the head section of the webpage. Multiple webpages can use the same link to access the style sheet.

This link to an external Stylesheet is placed within the head section of the HTML document.

example

<pre> .html (File) <head> <link rel="stylesheet" </head></pre>	Name of the CSS file written Here <pre>href="one.css"></pre>
---	--

<body>

<h1> External CSS </h1>

</body>

- We use link tag for connect CSS file with HTML document.
- "href" is used for specify the URL or name of CSS file.

one. CSS (File)

```
h1 {
    color: red;
}
```

→ Selectors

Selectors are the first part of CSS rule. It is a pattern of elements and other terms that tell the browser which HTML elements should be selected to have the CSS property values inside the rule applied to them.

The elements which are selected by the selectors are referred to as the subject of the selectors.

In CSS mainly four types of selectors are present.

- i) Element Selector
- ii) Class Selector
- iii) Id Selector
- iv) Group Selector

i) Element Selector

Element Selector selects HTML element based on the element name and the names are h1, p, form, ul, etc.

Syntax

element - name {
CSS property

example

<head>

<Style>

h1 {

color: green;

h1 }

Name of the
HTML element
are target here

{

color: red;

p }

</Style>

</head>

HTML Element

<body>

<h1> I am Heading (<h1> element </h1>
<p> I am Paragraph (<p> element </p>

<body>

ii) Class Selectors

For target a class first we put class attribute in our any HTML tag or element and gave it to special name. And Second we target that class in CSS File with the dot (.) Character.

Syntax

•> Class name set in HTML element or tag

{<h1 class="classname"> Content </h1>}

•> Target a class name with Dot in CSS

• class name {

CSS Property

}

example

```
<head>  <Style>
        .one {
            color : blue ;
        }
</Style>
</head>

<body>
    <h1 class = "one" > Name of class is
    one . </h1>
</body>
```

iii) id selectors

"id selectors" is same as class selector just a small change in this, we use (#) hash instead of dot (.) character and "id" instead of "class".

Syntax

→ 1st name set in HTML element

```
<h1 id = "id name" > Content </h1>
```

→ Target a id name with hash (#) in CSS.

```
#id name {
    # CSS property
}
```

example

<head>

<Style>

#one {

color: red;

}

</Style>

</head>

<body>

<h1 id = "one">My id name is one.</h1>

iv) Group Selectors

Group Selectors is mainly used for select more than one elements which we want to set the same style of definition or CSS. Basically it is minimize the code.

Syntax

→ without Group Selector

h1 {

CSS property

}

p {
 # CSS property
}

•> With Group Selector

h1, p {
 # CSS property
}

example

<head>

<style>

h1, h2, p {

color : blue ;

}

</style>

</head>

<body>

<h1>	9 am	Heading (1)	</h1>
<h2>	9 am	Heading (2)	</h2>
<p>	9 am	Paragraph	</p>

</body>

→ Comments

Like HTML, in CSS also comments are used. These comments are also ignored by the browser like HTML comments. It is used for tell the reader that CSS codes is used for that HTML document. And also it is used for leave notes like HTML comments.

Syntax

/* _____ Comment _____ */

example

<Style>

Comment



```
h1 { /* Hello am CSS comment */
    color : red ;
}
```

```
p {
    color : red ;
}
```

</Style>

→ Colors

Color property is mainly used for set the color of HTML element, color of fonts, background color, border color etc. It is also called decorative property.

Syntax

```
element {
    color : Color name ;
}
```

• There are mainly four ways for set a color.

- i) Keyword values
- ii) Hexadecimal values
- iii) RGB Value
- iv) HSL value

i) Keyword Value

In the keyword value you just write down a name of the color in the color property. For example, if we want red color of our heading then we just type the color name red.

example

Let's take an example of Keyword value, in this we set the "greenyellow" color of our heading.

<head>

<style>

h1 {

color: greenyellow;

</style>

</head>

<body>

<h1>

Heading

</h1>

</body>

ii) Hexadecimal value

Hexadecimal value can be defined as a six digit color representation. It starts with hash (#) symbol and followed by six characters range from 0 to 255 and A to F.

{ For example, # 00 00 00 }

In the example, of Hexadecimal value the first two digits representation. It starts with hash (#)

the red color. The next two digit represent the green color and the last two digit represent the blue color.

Example

Let's take an example of Hexadecimal value, in this we set the "red" color of our Heading, it is written in Hexadecimal value ~~# F00 700 7~~ ~~# F00 700 7~~.

```
<head > < style >
    h1 {
        color : F00707;
    }
</ style >
</ head >

< body >
    < h1 > Heading < h >
</ body >
```

iii) RGB value

The Full form of RGB is red, green and blue. And the range of RGB is 0 to any numbers. This property allows two values that can be either in % (Percentage) or any integer.

for example, rgb (30, 345, 123)

example

Let's take an example of RGB values, in this we set the "light green" color of our heading, it is written in RGB value in rgb "(30, 345, 123)".

```
<head>
  <style>
    h1 {
      color: rgb (30, 345, 123);
    }
  </style>
</head>

<body>
  <h1> Heading </h1>
</body>
```

iv) HSL Value

The Full Form of HSL value is Hue, Saturation, Lightness. Let's understand HSL in brief.

→ Here, it is mainly used for select the color from color wheel. The value of hue is in degree from 0 to 360.

→ Saturation means the visibility of color, its value is in % and the range of value is 0 to 100. And the main thing is that, with the increasing of Saturation percentage, the visibility of color is increasing.

→ Lightness is used for making glow the color. It's value is in % (Percentage) and the range of its value is any number. And with the increase of % the lightness is increased.

example

Let's take an example of HSL values, in this we set the "yellow" color of our heading, it is written in HSL value is `hsl (230, 45, 300)`.

`<head>`

`<style>`

`h1 {`

`color: hsl (230, 45, 300);`

`}`

`</style>`

`</head>`

`<body>`

`<h1>`

Heading

`</h1>`

`</body>`

→ Background

The Background property is used for the set background color of HTML document, background image of HTML document. And the repetition, attachment, position of the image.

Syntax

```
element {  
    background - any background property: value;  
}
```

At the place of any background property we just put the color, image, etc.

Let's study about background properties in brief.

i) background - color:-

The "background - color" property is used for set the background color of an HTML element. In the background - color property you set every type of color value like keyword value, RGB value etc.

example

```
<head>
  <script>
    h1 {
      background-color : red;
    }
  </script>
</head>

<body>
  <h1> Background color </h1>
</body>
```

ii) Background - image

The "background - image" property is used for set an image at the background of an element. By default, the image cover entire element but you set that image by the use of padding and margin which we will study in CSS padding margin.

example

```
<head>
  <script>
    body {
      background-image : url ("Pic.jpg");
    }
  </script> </head>
```

example

```
<body {  
    background - image : url ("Pic .jpg");  
    background - repeat : no - repeat ;  
}
```

Here you can set the value of repeat.

iv) background - attachment

The "Background - property" is mainly used for show the background image is scroll or Fixed when the content of the webpage is scroll.

- Fixed - In the fixed value, if an element has a scrolling mechanism then background does not move with the element.
- Local - In the local value, if an element has scrolling mechanism then background scrolls with the element content.
- Scroll - In the scroll value, the background is fixed relative to the element itself and does not scroll with its content.

```
<body>
  <h1> Background Image </h1>
</body>
```

iii) background - repeat

When you add an image by the use of background - image attribute, By default, this image will repeats vertically and horizontally. So, by the use of background - repeat property we set the image only horizontally or vertically and also we set the no-repeat if we want a image on webpage.

In the background - repeat we use three values for set the image, repeat-x, repeat-y, and no-repeat.

- > "repeat-x" is used for make image horizontally repeated.
- > "repeat-y" is used for make image vertically repeated.
- > "no-repeat" is used for show only single image, not repeated horizontally and vertically.

Syntax

```
body {
    background-image: url("Pic.jpg");
    background-attachment: Fixed, scroll;
}
```

v) background position

The "background-position" property is used for set the position of an image like the align attribute of HTML image. Here we just change a tag and the use of tag.

By default, the background image is placed on the top left on the webpage.

There are five Different values, which is used for set the position of image - top, bottom, center, left, right.

Syntax

```
body {
    background-image: url("Pic.jpg");
    background-attachment: Fixed;
    background-position: top, right - etc;
}
```

→ Border

Border is the important element in web design. They can be used for separate the content on a webpage, making it easier for people to understand and take action on your site. In simple words, it makes a line between two different contents for easily readable.

Border have mainly four Properties which is used for gave a style to border, color of border, set the size of border and the radius of border.

- i) border - style
- ii) border - color
- iii) border - width
- iv) border - radius

i) border - style

The "border - style" property is used for specify the type of border which you want to display on the screen. We have mainly nine different types of border.

These borders are —

example

<head>

<script>

- none { border - style : none ; }
- dotted { border - style : dotted ; }
- dashed { border - style : dashed ; }
- Solid { border - style : Solid ; }
- double { border - style : double ; }
- groove { border - style : groove ; }
- ridge { border - style : ridge ; }
- inset { border - style : inset ; }
- outset { border - style : outset ; }
- hidden { border - style : outset ; }

</script>

</head>

<body>

<p class = "none" >	None Border	<p>
<p class = "dotted" >	Dotted	<p>
<p class = "dashed" >	Dashed	<p>
<p class = "Solid" >	Solid	<p>
<p class = "double" >	Double	<p>
<p class = "groove" >	Groove	<p>
<p class = "ridge" >	Ridge	<p>
<p class = "inset" >	Inset	<p>
<p class = "outset" >	Outset	<p>
<p class = "hidden" >	Hidden	<p>

</body>

ii) border - color

"border - color" is used for set the color of borders. A choose exact border color for web pages content, make a webpage amazing.

Syntax

• any class, id, etc {

border - style : solid;

border - color : red;

}

iii) border - width

with exact border and border - color the third main thing is set in the web page is border - width. We set the width of border in pixel and also in predefined value like thin, thick and medium.

Example

< head >

< style >

h1 {

border - style : solid;

border - width : 4px;

}

width of Border in pixel

h2 {

border - style : solid ;

border - width : medium ;

}

Border width in
predefined value.

iv) border - radius

"border - radius" property is used for make corner rounds. For set the corners rounds we use four different types of values Single value, two value, three value and four values. And we set the values in px - (pixel) and % (percentage).

•> First value - In this we write down a one value "border - radius : 30px ;" which is set for all the corners.

•> Two value - In this we write down two values "border - radius : 20% 10% ;". The first value 20% is round the Top left and Bottom right corners. The second value 10% is round the top right and Bottom left corners.

→ Three value - In this we write down three values "border-radius: 10% 20% 30%;". The First value is round the to left corner. The Second value is round the top right and Bottom left corner. The third value is round the bottom right corner.

→ Four value - In this we write down four values "border-radius: 10% 20% 30% 40%;". The first value is round the top left corner. The second value is round the top right corner. The third value is round the bottom right corner. The fourth value is round the bottom left corner.

example

< head >

< script >

```
div {
padding : 60px;
margin : 18px;
border : 4px ridge red;
width : 350px;
float : left;
height : 170px;
```

Don't worry about it

```
}
```

```
p {
    font-size : 20px;
}
```

```
# div - one {
    border-radius : 30px;
    background : green;
}
```

```
# div - two {
    border-radius : 20% 10%;
    background : blue violet;
}
```

```
# div - three {
    border-radius : 10% 20% 30%;
    background : yellow;
}
```

```
# div - Four {
    border-radius : 10% 20% 30% 40%;
    background : Orange;
}
```

```
</script>
</head>
```

```
<body>
```

```
<div id = "div - one">
    <h2> First value </h2>
    <p> border-radius : 30px </p>
</div>
```

```
<div id="div - two">  
  <h2> Two values </h2>  
  <p> border-radius: 10% 20% </p>  
</div>
```

```
<div id="div - three">  
  <h2> Three values </h2>  
  <p> border-radius: 10% 20% 30% </p>  
</div>
```

```
<div id="div - Four">  
  <h2> Four values </h2>  
  <p> border-radius: 10% 20% 30% 40% </p>  
</div>
```

```
</body>
```

→ Font

Font is the main part of the websites or webpages, that's why choosing the right font can create a strong identify for your websites. Font property basically control the all structure of font like, size, style, color and etc.

In font family mainly these properties are used for make a font beautiful.

i) Font size

"Font-size" property is mainly used for adjust the size of font. In this we set the size of font in two different values. Keyword values and Adjustable values.

→ Keyword Values

In the keyword values, the size is fixed. By default we just gave values in keyword like large, xx-large etc.

In this mainly nine values are present, they are — smaller, small, x-small, xx-small, medium, large, larger, x-large, xx-large.

example

<body>

<p style = "Font-size : smaller ;"> Smaller </p>

<p style = "Font-size : medium ;"> Medium </p>

<p style = "Font-size : xx-large ;"> xx-Large </p>

</body>

→ Adjustable values

In the Adjustable value we get set a size of font is % (Percentage) and px (pixel). Here you can set the font size which you like or which size is suits perfectly to your webpages or sites.

example

```
<body>  
  <p style = "Font - size : 20% > Font Size <p>  
  <p style = "Font - size : 20px > Font Size <p>  
  
</body>
```

(i) Font - Style

"Font style" property is mainly used for set the style of the font which you want to display on the webpages. For set the style of font we mainly used four values bold, oblique, normal and italic.

example

```
<body>  
  
  <p style = "Font - style : italic;" > Italic <p>
```

iii) Font - weight

"Font - weight" is used for set the boldness of font or specifies that how bold is font. We have two different values for the boldness of fonts.

→ Keyword values

In the keyword values the boldness of font is set by that values whose boldness is already settle, like we have mainly three values bold, bolder, lighter, These values have a by default font weight.

example

```
<body>
```

```
<p style = "Font weight; bold;" > Bold </p>
```

```
</body>
```

→ Adjustable value

In the adjustable value we set the weight of font which we like or which suits our webpages on sites.

example

<body>

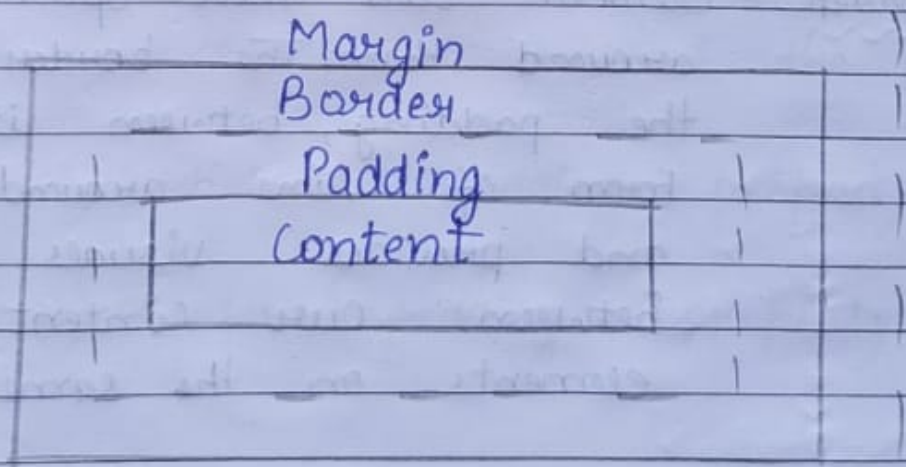
<p style = "Font - weight : 300 ; ">Weight <p>

</body>

→ Box Model

Box Model is chamber that includes content, padding, border and margin. These are mainly parts of making webpages; because it is used for develop the structure and design of the webpage. According to the box model browser supplies each element as a square prism.

Diagram



The Diagram of Box Model shows that any web page we see is made up of elements wrapped in rectangular boxes and arranged in relation to each other. These elements coexist above, besides, within each other, depending on the type of element they are:

Explanation about the parts of Box Model

Content - Every element starts with some content like text, image and etc. The content is placed inside a box that is just big enough to contain it.

Padding - Padding is used for clearing an area around the content. And it is transparent.

Border - Element can have optional border around them. The border surrounds the padding, between it takes the form of a line around our content and provides visual separation between our content and other elements on the same page.

Margin - Margin is used for clearing an area outside the border. And it is also transparent like padding.

→ Padding

Padding property describes the space between content and border. It is used for create visual white-space between the content and border of the box. It is transparent and has no color or decoration of its own.

Using CSS, you're going to be able to control the padding on any side (top, right, bottom, left).

Values for Padding properties

- Length - It's mean that the value of Padding is in px (pixel), cm (centimeter).
e.g. 10px, 10cm.
- % - It's mean that the value of Padding is in % (Percentage) .e.g. 10%, 20%.

We have two properties for set the padding.

- i) Individual padding property
- ii) Shorter padding = property

i) Individual Padding Property

In this we set the padding from each side (top, left, right, bottom) separately.

example

```
<head>
  <script>
    p {
      background-color: red;
    }
    .padding {
      padding-top : 50px;
      padding-bottom : 50px;
      padding-right : 50px;
      padding-left : 50px;
    }
  </script>
</head>
```

```
<body>
```

```
<p class = "padding"> Individual </p>
```

```
</body>
```

ii) Shorten Padding Property

In this we set the padding of each side in one property. We set the padding with ~~Four~~ Four different values - one value, two values, three values, Four values.

→ One value

`padding: 10px;` with a one value we set the padding from all side, here we declare 10px padding for all side content.

→ Two values

`"padding: 10px 20px;"` with a two value the padding of content divides in two parts, top and bottom, left and right. Here 10px is value for top and bottom padding, 20px is for left and right padding.

→ Three values

`"padding: 10px 20px 30px;"` with a three values the padding of content divides the three parts, Top padding, bottom padding and right-left padding. Here 10px is the value of top padding, 20px is the value of right and left padding, 30px is the value of bottom padding.

→ Four values

"padding : 10px 20px 30px 40px;" with a four value the padding of content divides into four parts. Top padding, Bottom padding, Left padding and Right Padding. Here 10px value of top padding, 20px is a value of right padding, 30px is value of bottom padding, and 40px is value of left padding.

example

<head>

<script>

p {

background-color: red;

.padding {

padding: 10px 20px 30px 40px;

</script>

</head>

<body>

<p class="padding"> Shorten Property </p>

</body>

→ Margin

Margin gives you away to add space between two elements which exists at same page, like if two boxes are present next to each other, then margin act as the space in between them.

Like padding margin are also transparent and have no color (of entire) or any decoration of their own.

Using CSS, you're going to be able to to control the width of entire margin or any particular side (top, right, bottom, left).

Values for Margin Properties

→ Auto

If two boxes present at browser then browser set automatically some margin between them.

→ Length

It's mean's that value of margin is in px (pixel), cm (centimeter) .e.g. 10px, 10cm.

→ %

It's means that the value of margin is %
e.g. 10%, 20%

→ Margin Property

```
<head>
  <script>
    p {
      background-color: red;
      .margin {
        margin-top: 30px;
        margin-right: 30px;
        margin-bottom: 30px;
        margin-left: 30px;
      }
    }
  </script>
</head>

<body>
  <p class="margin"> Margin </p>
</body>
```

→ Float

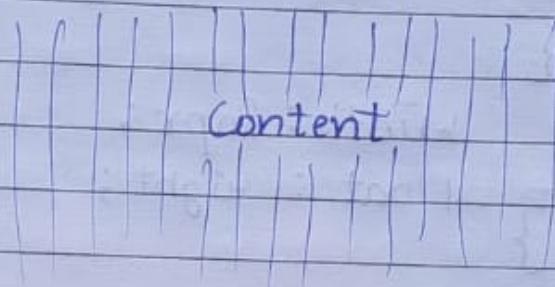
Float is the positioning property. It is used to push an element, image, layouts to the left side or right side at the browser. We can float anything with three easy steps.

Step - 1

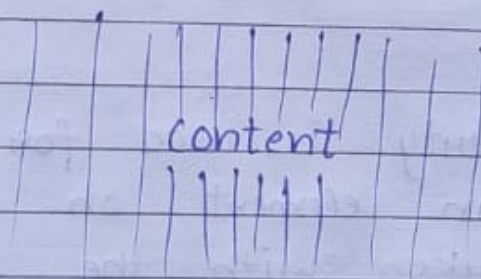
Let's take one paragraph which contain text and one image. Image is between anywhere in the text which we write in our paragraph tag.

example

< p >



< img src = "Pic . jpg" >



< / p >

Step - 2

A requirement for any Floating element is having width. So in the 2nd Step we set a width for image.

example

```
img {
    width: 80px;
}
```

Step - 3

Now let's add the Float property. The Float property can be set to either right or left. By default, the left float is already mainly when we want to float our element at right side.

example

```
img {
    width: 80px;
    float: right;
}
```

→ Position

Position Property is used for defines the position of an element on webpage. This is property works with the right, left, top, bottom properties to determine the final position of the element on a webpage.

The Position Property takes five values for the set the position - Static, relative, Fixed, absolute, Sticky.

i) Static

"Static" is the default position for HTML elements. Elements with position static are positioned based on the normal flow of the Page.

ii) Relative

"When we assign the "position: relative" on element follows the render flow of the page, but will be shift relative to its initial position.

iii) Fixed

Elements with "position: fixed" do not adhere to the normal render flow of the document. Fixed elements do not move when the user scrolls and they do not leave a blank space where the element would have been positioned.

absolute

iv) With "position: absolute", an element ignores the normal document flow, if you give an element position absolute all other elements will act as if the absolute positioned element doesn't even exist.

v) Sticky

Elements with "position: sticky" are positioned depending on the user's scroll. It is depending on how far the user can scroll the page, a sticky element ^{behaves} like relative element until the viewport ~~web~~ meets a specified position.

Then it becomes fixed a spot and appears to "stick" to the page as the user scrolls.

example → Let's take example for "Fixed" position

<head>

<script>

one {

background-color: red;

}

two {

position: Fixed;

top: 28px;

left: 28px;

background-color: green;

}

three {

background-color: orange;

}

```
div {
    width : 70px;
    height : 70px;
    margin : 10px;
    border-radius : 8px;
    text-align : center;
    line-height : 70px;
    color : white;
    font-size : 25px;
}
```

```
</script>
</head>
```

```
<body>
```

```
<div id = "one" > 1 </div>
<div id = "two" > 2 </div>
<div id = "three" > 3 </div>
```

→ Overflow

Overflow occurs when element content does not fit inside the element box. This can happen when an element has specified height that's small for contain the content.

According to the box Model, A box is generated for HTML elements. The box comprises four layers Content, Padding, Border and Margin.

In this when the content of HTML document elements extends beyond any of edges, whether that's the content edge, padding edge, border edge and margin edge. It is called overflow.

How Solve Overflow Problem?

We have two ways for solving the overflow problem.

- i) The first way is that reduce the content which you write the box make that size of box which easily contain all the content.
- ii) The Second way, if your box 'size is OKay' for your website or webpages and you can't reduce your content then you can use overflow scroll property.

→ Overflow Scroll

To hide overflow from rendering outside the element's box while enabling users to view that content. set the overflow property to "scroll". With this property all the content are located inside the element's box, for visibility of this content you can just scroll up and down.

example

<head>

<script>

div {

overflow : visible ;

box - sizing : border - box ;

width : 320px ;

height : 200 px ;

padding : 20 px ;

border : 1px Solid ;

background - color : yellow ;

margin : 10 px ;

border - radius : 10px ;

</script>

</head>

<body>

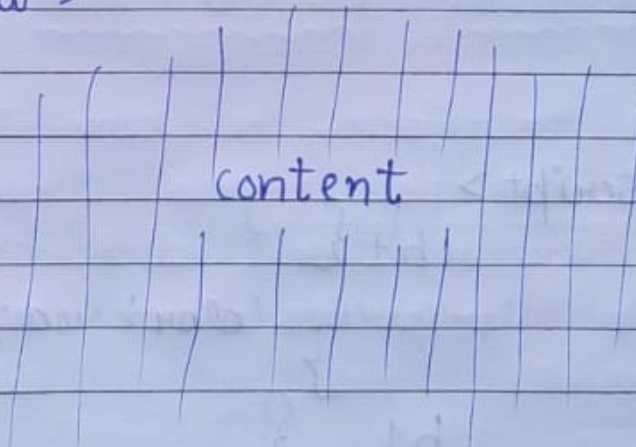
<div>

content

</div>

</body>

Visible show the overflow if
you change it with scroll
then it changes in scroll



→ !important

Important means "this is important". It means that only the !important property value is to be applied on the element and all the other properties value are to be ignored.

In other words, an important property is used for override the other same styling properties.

Suppose we take one heading. First we set the color of heading is red, and also we gave it the blue color by writing one more value with !important value. Then we get our heading in blue color. Because blue color get higher priority as compare to the red color by the use of !important value.

example

```
<head>
  <script>
    h1 {
      color: red;
    }
    h1 {
      color: blue !important;
    }
  </script>
</head>
```

```
< body >
    < h1 > Important < h1 >
< /body >
```

→ Hover Effect

Hover animation occurs when a user hovers an element with their cursor and the element responds with any animated effect. In simple words, hover effect means the mouse over effect or animation. Like if we made a normal text or add any image, we want some effect like color changing effect on text and zoom effect on images when our mouse will over on it. Then we used image magic of hover effect which give the color changing and zoom effect give the color ch to our text and images when our mouse will over on it.

Syntax

```
: hover {
    Decoration Property
}
```

we show : hover effect in our coming transition part.

→ Transform Property

Transform property changes the position, shape and the size of the element. A CSS transform rule is written as follows.

Transform value (arguments);

Here, value is the type of transformation and argument is the extent to which the transformation is applied.

CSS transform can be split into two categories.

→ 2-Dimensional Transformation

→ 3-Dimensional Transformation

→ 2-Dimensional Transformation

In 2-D transform, the element transform in two axes. i.e. X-axis (horizontal) and Y-axis (vertical). We used mainly five transform properties for make 2D Transform.

↓ Properties and Use of Properties

i) Translate () property

We used translate property is for move an element from one place to another place in x-axis and y-axis.

ii) Scale () property

We used scale property to scales up and down an element by the increasing and decreasing the height and width of the element.

iii) Skew () property

We used skew property to skew or slant the element along x-axis and y-axis.

iv) rotate () property

We used rotate property for rotate an element.

v) matrix () property

Matrix property is the master property. In this you set all the upper properties in once.

→ 3-Dimensional Transformation

In the 3D transform, the element transform in three axes, i.e. X-axis, Y-axis, Z-axis. We used mainly these properties for make 3D transform.

↓ Properties and Use of Properties

i) rotate () property

We used rotate property for rotate an element along X-axis, Y-axis, Z-axis.

ii) perspective ()

In a simple words perspective is a distance between the computer screen and your eyes.

iii) translate () property

We used translate property is for move an element from one place to another place in X-axis, Y-axis, Z-axis.

→ Transition Property

In last section, we learn about Transform property, and we study that transform is used for move or change the appearance of an element.

Now we will study about transition, it is same as like transform but a small different. Transition makes the element smoothly and gradually change from one position to another position. It works like grease in the wheel,

without a transition, an element being transformed would change abruptly from one position to another position. By applying a transition you can control the change, gradual, and make it very smooth.

For creating transition we want specify two things.

- The CSS property on which the element you want to add an effect.
- Time duration of the effect.

Let's take an example of transition effect on element with time duration of 5 seconds and the width property.

> Step - 1

First we take div tag in our HTML file.

```
<div> _____ </div>
```

> Step - 2

In a second step, we set the height and width of the div tag and also set the color of the div tag.

```
div {
    height : 100px;
    width : 100px;
    color : red;
    -webkit-transition : width 25;
    transition : width 25;
}
```

> Step - 3

After set the transition on div tag we set : hover property at width of 250px.

```
div : hover {
    width : 250px
}
```

→ Animation

Animation is the movements changes and actions of the elements on a webpage. They make a site live, lively and dynamic. It adds the better user experience to a page since we human tends to interact better with a dynamic environment. It is also add the individuality and personality to a web visual.

CSS transition property already takes care of the changes and the movements on webpages, then,

Why we need Animation???

- Transition is simple change while changes in animation can be more complex.
- Animation allows you to gain more content over the process.
- Animation can loop, while transition can not.
- Animation is flexible and dynamic.
- Transitions are made with the three distinct area - start, end and duration. But the animation is free from the dependency of any state - they can come into play whenever.

Basic of Animation

Animation requires three basic things.

→ Key Frames

It is define the style or appearance of the element and must be defined.

→ Properties

These are the animation properties attached the key frames.

→ Duration

The Duration (time) of the entire animation process.

Properties Associated with Animation and @ Keyframe

i) animation - name

It defines the name of the @ Key frames animation, The name is like variables and it up to the developer to specify.

ii) animation - duration

It is defining the time it takes for the animation to complete one cycle. The value for this is an integer in seconds and the minimum value is 0 sec.

iii) animation - delay

It is time delay between the element load time and the beginning of the animation sequence.

iv) animation - iteration - count

It defines the number of times an animation should be repeated.

v) animation - direction

It defines the direction of animation left, right or alternate.

vi) animation - timing - function

It defines the time or speed at which the animation changes from one style to another style. It has the values as linear, ease-out, ease-in, step-start, step-end, and cubic-bezier.

vii) animation shorthand

"animation" is the shorthand property for all the sub properties which we read before this property. This is also called master property with this you can set the time, name, etc.

How it works ???

One a keyframe is defines, the animation sequence is configured in % (Percentage) to specify the start and end progress of the animation. 0% is usually the start value and 100% the end. In between 0 and 100, intermediate stages can be specifies, like 30%, 75%, etc. telling the browser what to do at each stage.

you can also replace the 0 to 100 by "From" and "to".

Let's take an example, In this example we make a color changer without any stop with use of Animation.

Here is three steps for making animation

→ Step - 1

In a step one, first we take a div tag in our HTML file.

```
<div></div>
```

→ Step - 2

In a second step, we set the height and width of the div tag on animation screen, we set it 100% for cover full screen. After that we gave a name of the animation is color and time for color changing animation is 6s, means 6 second and the animation - thing function at infinite.

```
div {
    height : 100% ;
    width : 100% ;
    animation : color 6s infinite ;
}
```

→ Step - 3

In a third step, we set @ keyframe rule with animation name and set the value 0%, 50%, 100% and color for all that values.

@ Keyframes color {

```
0% {  
    background-color: green;  
}  
50% {  
    background-color: red;  
}  
100% {  
    background-color: orange;  
}
```

→ Responsive Design

In May - 2010 Ethan Marcotte describes the Responsive Design in his article. In this he mainly describes the Flexible images, media queries and fluid grid.

Responsive web design is basically used for resize, hide, shrink, move the content of the website for looking good when its screen size will be small, big, medium, which is fit in Desktop, Tablet and Mobile Phones.

Before mobiles phones web pages were designed only for computer screen, But now a time daily new device are invented. So, we can't possibly design a separate website or webpages for each potential device your visitors may use.

That's why we use Responsive Design for making a webpages and websites, because it is compatible with all devices and screen to ensure a delightful experience, both modern day devices and those yet to be invented.

→ View port

View port is the area of webpage which is visible to the user. The viewport is fundamental to web development, since its dimensions are what control how page elements appear.

On the developer's desktop device, the viewport size matches the browser window size, and on a mobile device the view port is generally the size of the devices screen.

There are two types of "viewport", the layout and the visual.

- "The Layout" viewport refers to the dimensions of the page.
- "The visual" viewport is most relevant on mobile devices, where these browser events are most common like - one Screen Keyboard Zoom, events, etc.

```
{  
  p {  
    }  
  }  
} Font-size: 18px;
```

Let's break the code for understanding how @ media rule is used for making responsive viewport.

- First, we take a paragraph in our HTML which contains some content.
- Second, we declare the font size for content which is inside the paragraph to be 16px.
- Third, but the first part of @ media rule. Here @ media rule is the media type, which tells the browser which kinds of pages to apply the styling like in our example we choose screen.
- In the second part of @ media rule is, (max-width: 400px;) we declare that styling should be only apply to a screen with a maximum width of 400px.
- The final part of the @ media rule is the alternative styling. In this we gave alternative (font-size: 18px;) for our paragraph when they open in small device.