Syllabus

Introduction of css

Background

-how to setup

Box-Model

-what is box model in css

-how to work.

Selectors:

what is selector ?

types of selector

how to apply selector

Combinators

what is Combinators

types of Combinators

how it is related to Selectors

Pseudo-class & pseudo-elements

-what is Pseudo-class

-what is pseudo-elements

-types

-how to apply

Transition , transform and animation in css

- Application of Transition , transform and animation.

- use of Transition , transform and animation.

One Basic Project using html and css.

CSS

what is CSS

The full form of CSS is “cascading style sheet”.

We are using CSS to apply design in our web pages.

CSS are case-insensitive.

CSS is used to apply styles or design for our web pages and these are the

different types.

1) inline css // style="color=red ;"

2) internal css : inside the style tag and in the same html document styles are created

3) external css : syles are created in differt file with .css as an extension.

Background

It is used to apply the background for the entire web page or for one particular content.

>We can apply image or color as a background.

--background-color: gold; to apply color AS THE BACKGROUND

--background : url(images/image_name.jpg); to apply image AS THE BACKGROUND

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: margins, borders, padding, and the actual content.

We can create box model by using ,

--border properties.

--Outline properties.

We can have spacing using ,

--Padding properties

--Margin properties

Psuedo class:

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

Style an element when a user mouse over it

Style visited and unvisited links differently

Style an element when it gets focus

Types :

: Active

: Focus

: Hover

: First child

: Last child

Psuedo element :

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

Style the first letter, or line, of an element

Insert content before, or after, the content of an element

Types :

::selection Selects the portion of an element that is selected by a user

::after Insert content after every <tag> element

::before Insert content before every <tag> element

::first-letter Selects the first letter of every <tag> element

::first-line Selects the first line of every <tag> element

Display properties :

The display property specifies the display behavior (the type of rendering box) of an element.In HTML, the default display property value is taken from the HTML specifications or from the browser/user default style sheet. The default value in XML is inline, including SVG elements.

each and every tag will be having some default display properties.

ex: for h1 block is the display property.

for span inline is the display property.

1) block: it will take a new line to display its content and once after it will move the cursour to next line

2) inline : it will not take any other extra space except to display its content.

3) flex : Displays an element as a block-level flex container

4) grid : Displays an element as a block-level grid container.

5) 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) inline-flex : Displays an element as an inline-level flex container

Transisation

Transition : to apply the stylings based on the time function we use transition.

transition-property Specifies the name of the CSS property the transition effect is for

transition-duration Specifies how many seconds or milliseconds the transition effect takes to complete

transition-timing-function Specifies the speed curve of the transition effect

transition-delay Defines when the transition effect will start

Transform :The transform property applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, move, skew, etc., elements.

scale(x,y) Defines a 2D scale transformation

scale3d(x,y,z) Defines a 3D scale transformation

scaleX(x) Defines a scale transformation by giving a value for the X-axis

scaleY(y) Defines a scale transformation by giving a value for the Y-axis

scaleZ(z) Defines a 3D scale transformation by giving a value for the Z-axis

rotate(angle) Defines a 2D rotation, the angle is specified in the parameter

rotate3d(x,y,z,angle) Defines a 3D rotation

rotateX(angle) Defines a 3D rotation along the X-axis

rotateY(angle) Defines a 3D rotation along the Y-axis

rotateZ(angle) Defines a 3D rotation along the Z-axis

skew(x-angle,y-angle) Defines a 2D skew transformation along the X- and the Y-axis

skewX(angle) Defines a 2D skew transformation along the X-axis

skewY(angle) Defines a 2D skew transformation along the Y-axis

Animation

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times as you want.

To use CSS animation, you must first specify some keyframes for the animation.

Keyframes hold what styles the element will have at certain times.

RULES :

The animation-duration property defines how long an animation should take to complete. If the animation-duration property is not specified, no animation will occur, because the default value is 0s (0 seconds).

In the example above we have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).

It is also possible to use percent. By using percent, you can add as many style changes as you like.

The following example will change the background-color of the element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

------ END OF CSS ------