**\*CSS 3 Tutorial\***

**>>>By MAROOO<<<**

**=> prefixes:**

**That like a command, made for old browsers to understand modern codes**

**Code >> border:1px groove yellow;**

**Code with prefixes:**

**For google chrome > -webkit- border:1px groove yellow;**

**For moz firefox > -moz- border:1px groove yellow;**

**For opera > -o- border:1px groove yellow;**

**=> border radius:**

**for make the style like circle**

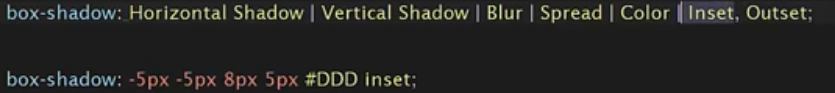
**Code => border-radius:10px;**

**For special side code => border-top-right-radius:10px;**

**border-bottom-left-radius:10px;**

**=> box shadow:**

**for make shadow for ur div as circle or square**

**code:**

**horizontal shadow: for left & right // vertical shadow: for up & down**

**=> Text Shadow:**

**Same things for box shadow but code change form**

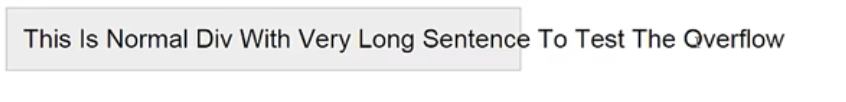
**Box-shadow: >>>> text-shadow:**

**With same attributes**

**=> Text-overflow:**

**To control in text which in div if width of div isn’t enough for text**

**Ex:**

****

**To solve this you can use overflow features but you will find other problem and it’s text no completed, you can fix it with text-overflow**

**1) Text-overflow: hidden;**

**To hide the text out of div**

**2) Text-overflow: ellipsis;**

**To hide the text out of div and leave a mark to tell user that text have more**

**=> 2D transform**

**>transform:rotate(Ndeg)**

**With this code you can change degree of the shape**

**Code > transform:rotate(90deg)**

**transform:rotate(-90deg)**

**>transform: Scale(width , height);**

**To change the size of div (width and height)**

**Code > transform: scale(width , height )**

**Transform: scale(2 , 2)**

**You can write one number if you want to scale same for the wid & hei**

**Transform: scale(2 , 2) = Transform: scale(2)**

**You can use code**

**Transform: scaleX(for width)**

**Transform: scaleY(for height)**

**>transform: translate(R&L,T&B);**

**to change the place of div (mkano)**

**code > transform: translate(40px , 40px); {40px right & 40px bottom}**

**transform: translate(-40px , -40px); {40px left & 40px top}**

**>transform: skew(R&L,T&B);**

**It’s like rotate but skew change the view degree**

**Code > transform: skew(30deg, 30deg); {30deg right & 30deg bottom}**

**transform: skew(-30deg, -30deg); {30deg left & 30deg top}**

**>transform: matrix;**

**you can use all last codes in one code**

****

**photo is described what matrix is do**

**you can do rotate with matrix but it’s required more something no need to use it**

**=> 3D transform**

**>transform:rotate(x/y/z)(Ndeg)**

**3D rotate can do like skew in 2D**

**Code > transform:rotateX(90deg)**

**transform:rotateY(-90deg)**

**>transform: translate(R&L,T&B,N&F);**

**to change the place of div (mkano)**

**code > transform: translate(40px , 40px,40px); {right&bottom &near}**

**transform: translate(-40px , -40px, -40px ); {left &top&far}**

**new in translate 3D the third attribute it’s make div come near or go far form user**

**3D Transform – perspective**

**The distance between my eyes and the view**

**Code perspective:50px;**

**3D Transform - Transform Origin**

**The return point for change the shape**

**Code transform-origin: top left;**

**3D Transform - Transform style**

**To save the 3D option for every item**

**Code : transform-style: preserve-3d;**

**3D Transform – backface-visiblity**

**to hide the back of item to show other item when user hover**

**code : backface-visibility: hidden;**

**> Background new feature <**

**1) background-size**

**To change photo size to div size**

**Code: background: (width)px height(px);**

**Opacity:**

**for make the item light percent**

**you can replace visibility:hidden; with opacity:0;**

**code opacity: (0 to 1);**

**\*Transition\***

**Transition features**

**1)property:**

**which elements you want to do transition on it**

**Code => transition-property:all;**

**Default value = all == width, background, all elements**

**Property values => (1) all > transitions for all elements**

**(2) none > disable all transitions**

**(3) all elements like width height etc.**

**For multi property**

**Code => transition-property: width, background;**

**/\* you cannot do good transitions with property only you should use transition-duration with property too\*/**

**2)Duration:**

**The time took to do transition**

**Code=> transition-property:width;**

**transition-duration: 1s;**

**You can do mutli time duration**

**Code => transition-property:width, background;**

**transition-duration: 1s, 2s;**

**example:**

****

**3)timing-function:**

**Style for do transition**

**Code=> transition-timing-function:ease;**

**Css have 5 style for do the transition:**

**1)ease 2)linear**

**3)ease-in 4)ease-out**

**5)ease-in-out**

**You will mostly use ease (no need more)**

**4)delay:**

**Time before do the transition**

**Code=> transition-delay: 3s;**

****

**Short Code for all transition elements:**

**\*Columns:**

**To divide the division into columns**

**Code 1 => Column-count:44;**

**For how many columns you want to divide the div**

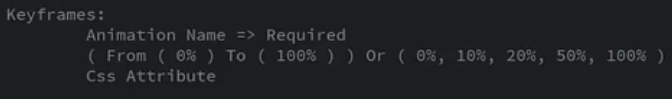
**Code 2 => column-gap:4px;**

**The space between columns**

**Code 3 => column-rule: 2px groove red;**

**To add border or vertical line between columns**

**\*Animation with CSS 3\***

**1) keyframe syntax:**

**2) Animation name:**

**Any animation have name and to make it work you should call it with his name**

**Code => animation-name: maro;**

**3) Animation duration:**

**The time took to do animation**

**Code => animation-duration: 5s;**

**4) Animation Iteration Count:**

**For how many time you want to do that animation**

**Code=> animation-** **iteration-count: 2;**

**You can write number for how many times you want do it**

**Or you can make it infinite for do it always**

**Code=> animation-** **iteration-count: infinite;**

**5) Animation Direction:**

**For the direction or the way to do animation**

**Code => animation-direction: D ;**

**D- 1) normal => for do it form first to last animation**

**D- 2) reverse => for do it form last to first animation**

**D-3) alternate => start for first to last animation , and we reach last**

**animation start again but form last to first animation**

**D-4) alternate-reverse =>**

**start for first to last animation , and we reach last**

**animation start again but form last to first animation**

**6) Animation delay:**

**time before start do animation**

**code=> animation-delay: 1s;**

**7) Animation fill mode:**

**The style after the animation end**

**Code=> animation-fill-mode: s ;**

**s- 1) none => default value for fill mode ;**

**s- 2) forwards => for end the animation at last keyframe you design it**

**s- 3) backwards => take first keyframe before start do animation**

**s- 4) both => for do forwards and backwards**

**8) Animation play state:**

**Code => animation-play-state: running;**

**animation-play-state: pause;**

**9) Animation timing function:**

**Style for do animation**

**Code=> animation-timing-function:ease;**

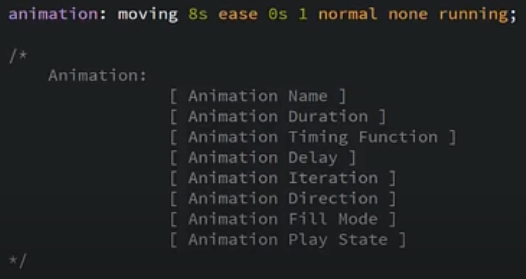
**Css have 5 style for do animaiton:**

**1)ease 2)linear**

**3)ease-in 4)ease-out**

**5)ease-in-out**

**You will mostly use ease (no need more)**

**=> short hand code for animation**

**\*more tips\***

**1) icons**

**You can use it with font awesome by add the stylesheet in heading and take the code form web site for them**

**2) background Gradients**

**That elements used to do color mix in background**

**Code => background-image: linear-gradient(details);**

1. **Color form top to bottom (default value) :**

background-image: linear-gradient(red, yellow);

1. **Color form left to right :**

background-image: linear-gradient(to right, red , yellow);

1. **Color form top left to bottom right:**

background-image: linear-gradient(to bottom right, red, yellow);