CSS Animations

# Transitions

transition is a css property that allows to change other properties of an html element smoothly over a given duration.

like doing something on hover etc.

button:hover {

    background-color: green;

}

but the above code makes the change of color happen so fast that its like on and off switch. transition is not smooth.

For transition we need two things:

1. css property which will be changed
2. duration – amount of time the above property change will take to complete.

As you will see blow we added transition property (1) and duration (2) to the button. now the change on hover happens over 1s smoothly.

button {

    height: 90px;

    width: 290px;

    font-size: 28px;

    color: white;

    border: none;

    background-color: blue;

    transition-property: background-color;

    transition-duration: 1s;

}

button:hover {

    background-color: green;

}

Note: we can also give duration in ms like 3000ms.

## Transition options

besides the above 2 properties required we have 2 more properties

1. transition timing function
2. transition delay

### Transition timing function

it defines how the change in property will occur. example if on hover the bg color changes at equal rate or changes rapidly first then slowly. it has 5 values:

1. ease: this is default value. change starts slow, goes faster, ends slow
2. linear: change occurs at same rate
3. ease-in: change starts slow and speeds up in the end
4. ease-out: change starts fast and slows in the end
5. ease-in-out: start slow, speed up, ends slow. similar to ease.

.move {

    margin: 2em;

    width: 150px;

    height: 50px;

    padding: 10px 15px;

    background-color: red;

    color: #ffffff;

    text-align: center;

    line-height: 50px;

    border-radius: .5em;

    font-size: 20px;

    transition-property: translate();

    transition-duration: 3s;

}

here transition is applied on translate property and duration is 3s

we can define timing function as follows

#linear {

    transition-timing-function: linear;

}

same for ease-in etc.

.move:hover {

    transform: translate(900px, 0);

}

on hover we translate 900px on x axis and 0 on y axis.

### Transition delay

time to wait before start of transition.

accepts time in seconds and milliseconds

In the button bg color change example if we add

    transition-property: background-color;

    transition-duration: 3000ms;

    transition-delay: 2s;

then the color change on hover starts after 2s and the change takes 3s as per duration given.

## Applying transition to different properties of the same element

just keep adding properties space separated to transition property.

**NOTE: ON SPECIFIC CHROME VERSIONS THE ANIMATIONS HAPPEN ON LOAD OF PAGE, TO AVOID THAT ADD TRANSITION PROPERTIES TO THE ACTION LIKE HOVER ETC.**

button {

    height: 90px;

    width: 290px;

    font-size: 28px;

    color: white;

    border: none;

    background-color: blue;

    /\* transition-property: background-color color width;

    transition-duration: 3000ms;

    transition-delay: 2s; \*/

}

button:hover {

    transition-property: background-color color width;

    transition-duration: 3000ms;

    transition-delay: 2s;

    background-color: green;

    color: black;

    width: 600px;

}

instead of using individual properties we can also define ‘all’ for transitioning all property changes.

    transition-property: all;

    transition-duration: 3000ms;

    transition-delay: 2s;

**Shorthand to write transition duration and property**

    transition: background-color 3s ease-in-out 1s,

    color 2s ease 4s;

note here that color change starts after background color has changed (1s delay and 3s duration)

1st param is propert, 2nd is duration 3rd is timing function and 4th is delay. we can have multiple comma separated properties.

**NOTE: to get oval shape use border-radius 50%**