CSS GRAPHICS

CSS Transforms

- CSS transforms allow you to translate, rotate, scale, and skew elements.
- CSS supports 2D and 3D transformations.

CSS 2D Transforms

- translate(): Moves an element from its current position
- rotate(): Rotates an element clockwise or counter-clockwise according to a given degree.
- scale(): Increases or decreases the size of an element
- scaleX(): method increases or decreases the width of an element.
- scaleY(): method increases or decreases the height of an element.
- skewX(): Skews an element along the X-axis by the given angle.
- skewY(): Skews an element along the Y-axis by the given angle.
- skew(): method skews an element along the X and Y-axis by the given angles.
- matrix(): Combines all the 2D transform methods into one.

matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())

```
transform: matrix(1, -0.3, 0, 1, 0, 0);
```

CSS 3D Transforms

- rotateX(): Rotates an element around its X-axis at a given degree
- rotateY(): Rotates an element around its Y-axis at a given degree
- rotateZ(): Rotates an element around its Z-axis at a given degree

CSS Transitions

- CSS transitions allows you to change property values smoothly (from one value to another), over a given duration.
- **transition:** A shorthand property for setting the four transition properties into a single property
- **transition-delay:** Specifies a delay (in seconds) for the transition effect
- **transition-duration:** Specifies how many seconds or milliseconds a transition effect takes to complete
- **transition-property:** Specifies the name of the CSS property the transition effect is for
- **transition-timing-function:** Specifies the speed curve of the transition effect

transition

- The transition property is a shorthand property for:
  - *transition-property*
  - *transition-duration*
  - *transition-timing-function*
  - *transition-delay*
- **Syntax:** *transition: property duration timing-function delay|initial|inherit;*
- **Eg:** *transition: width .35s ease-in-out 1s;*

transition-delay

- The transition-delay property specifies when the transition effect will start.
- Syntax: *transition-delay: time|initial|inherit;*
- Eg: *transition-delay: 2s;*

  - *Time:* *Specifies how many seconds or milliseconds a transition effect takes to complete. Default value is 0s, meaning there will be no effect*
  - *Initial:* *Sets this property to its default value.*
  - *Inherit:* *Inherits this property from its parent element.*

transition-duration

- The transition-duration property specifies how many seconds (s) or milliseconds (ms) a transition effect takes to complete.
- Syntax: *transition-duration: time|initial|inherit;*
- Eg: *transition-duration : 2s;*

transition-property

- The transition-property property specifies the name of the CSS property the transition effect is for.
- A transition effect could typically occur when a user hover over an element.
- **Syntax**: *transition-property: none|all|property|initial|inherit;*
  - ***None:*** *No property will get a transition effect*
  - ***All:*** *Default value. All properties will get a transition effect*
  - ***Property:*** *Defines a comma separated list of CSS property names the transition effect is for*
  - ***Initial:*** *Sets this property to its default value.*
  - ***Inherit:*** *Inherits this property from its parent element.*

Eg: *transition-property: width, height;*

Note: Always specify the transition-duration property, otherwise the duration is 0, and the transition will have no effect.

transition-timing-function

- The transition-timing-function property specifies the speed curve of the transition effect.
- Syntax: *transition-timing-function: linear|ease|ease-in|ease-out|ease-in-out|step-start|step-end|steps(int,start|end)|cubic-bezier(n,n,n,n)|initial|inherit;*
  - ***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*
- *Eg:* transition-timing-function: linear;

CSS Animations

- An animation lets an element gradually change from one style to another.
- You can change as many CSS properties you want, as many times 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.

Properties

- **@keyframes:** Specifies the animation code
- **animation:** A shorthand property for setting all the animation properties
- **animation-delay:** Specifies a delay for the start of an animation
- **animation-direction:** Specifies whether an animation should be played forwards, backwards or in alternate cycles
- **animation-duration:** Specifies how long time an animation should take to complete one cycle
- **animation-fill-mode:** Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)
- **animation-iteration-count:** Specifies the number of times an animation should be played
- **animation-name:** Specifies the name of the @keyframes animation
- **animation-play-state:** Specifies whether the animation is running or paused
- **animation-timing-function:** Specifies the speed curve of the animation

The @keyframes Rule

- When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.
- The @keyframes rule specifies the animation code.
- To get an animation to work, you must bind the animation to an element.
- The style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).
- **Syntax:** *@keyframes animationname {keyframes-selector {css-styles;}}*
- @keyframes mymove {

```
                from {top: 0px;}
                to {top: 200px;}

      }
```

CSS animation-direction Property

- The animation-direction property defines whether an animation should be played forwards, backwards or in alternate cycles.
- **Syntax:** animation-direction: normal|reverse|alternate|alternate-reverse|initial|inherit;
  - *Normal: Default value. The animation is played as normal (forwards)*
  - *Reverse: The animation is played in reverse direction (backwards)*
  - *Alternate: The animation is played forwards first, then backwards*
  - *alternate-reverse: The animation is played backwards first, then forwards*

CSS animation-fill-mode Property

- The animation-fill-mode property specifies a style for the element when the animation is not playing.
- Syntax: *animation-fill-mode: none|forwards|backwards|both|initial|inherit;*
  - ***None:*** *Default value. Animation will not apply any styles to the element before or after it is executing*
  - ***Forwards:*** *The element will retain the style values that is set by the last keyframe (depends on animation-direction and animation-iteration-count)*
  - ***Backwards:*** *The element will get the style values that is set by the first keyframe (depends on animation-direction), and retain this during the animation-delay period*
  - ***Both:*** *The animation will follow the rules for both forwards and backwards, extending the animation properties in both directions*

CSS animation-play-state Property

- The animation-play-state property specifies whether the animation is running or paused.
- Syntax: *animation-play-state: paused|running|initial|inherit;*
  - *Paused: Specifies that the animation is paused*
  - *Running: Default value. Specifies that the animation is running*