SHADOWS & ANIMATIONS

By Saddam Khan

Content:-

- Box-Shadows.
- Key Frames.
- Animations.
- Pseudo Classes.
- BackGround.
- Gradients.

box-shadow

The `box-shadow` CSS property adds shadow effects around an element's frame. You can set multiple effects separated by commas. A box shadow is described by X and Y offsets relative to the element, blur and spread radius, and color.

Specify a single box-shadow using:

- Two, three, or four `<length>` values.
  - If only two values are given, they are interpreted as `<offset-x>` and `<offset-y>` values.
  - If a third value is given, it is interpreted as a `<blur-radius>`.
  - If a fourth value is given, it is interpreted as a `<spread-radius>`.
- Optionally, the `inset` keyword.
- Optionally, a `<color>` value.

To specify multiple shadows, provide a comma-separated list of shadows.

```
/* Three length values and a color */
/* <length> | <length> | <length> | <color> */
box-shadow: 10px 5px 5px black;

/* Four length values and a color */
/* <length> | <length> | <length> | <length> | <color> */
box-shadow: 2px 2px 2px 1px rgb(0 0 0 / 20%);

/* inset, length values, and a color */
/* <inset> | <length> | <length> | <color> */
box-shadow: inset 5em 1em gold;

/* Any number of shadows, separated by commas */
box-shadow:
  3px 3px red inset,
  -1em 0 0.4em olive;

/* Global values */
box-shadow: inherit;
box-shadow: initial;
box-shadow: revert;
box-shadow: revert-layer;
box-shadow: unset;
```

@keyframes

The `@keyframes` CSS at-rule controls the intermediate steps in a CSS animation sequence by defining styles for keyframes (or waypoints) along the animation sequence. This gives more control over the intermediate steps of the animation sequence than transitions.

Definition and Usage

The `@keyframes` rule specifies the animation code.

The animation is created by gradually changing from one set of CSS styles to another.

During the animation, you can change the set of CSS styles many times.

Specify when the style change will happen in percent, or with the keywords "from" and "to", which is the same as 0% and 100%. 0% is the beginning of the animation, 100% is when the animation is complete.

Tip: For best browser support, you should always define both the 0% and the 100% selectors.

Note: Use the animation properties to control the appearance of the animation, and also to bind the animation to selectors.

Note: The !important rule is ignored in a keyframe (See last example on this page).

```
div {
  width: 100px;
  height: 100px;
  background: red;
  position: relative;
  animation: mymove 5s infinite;
}

@keyframes mymove {
  from {top: 0px;}
  to {top: 200px;}
}
</style>
</head>
```

CSS Syntax

```
@keyframes animationname {keyframes-selector {css-styles;}}
```

Property Values

Value	Description
animationname	Required. Defines the name of the animation.
keyframes-selector	Required. Percentage of the animation duration. Legal values: 0-100% from (same as 0%) to (same as 100%) **Note:** You can have many keyframes-selectors in one animation.
css-styles	Required. One or more legal CSS style properties

animation

The `animation` shorthand CSS property applies an animation between styles. It is a shorthand for `animation-name`, `animation-duration`, `animation-timing-function`, `animation-delay`, `animation-iteration-count`, `animation-direction`, `animation-fill-mode`, `animation-play-state`, and `animation-timeline`.

Syntax

CSS

```
/* @keyframes duration | easing-function | delay |
iteration-count | direction | fill-mode | play-state | name */
animation: 3s ease-in 1s 2 reverse both paused slidein;

/* @keyframes duration | easing-function | delay | name */
animation: 3s linear 1s slidein;

/* two animations */
animation:
  3s linear slidein,
  3s ease-out 5s slideout;
```

What are Pseudo-classes?

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 mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

Syntax

The syntax of pseudo-classes:

```
selector:pseudo-class {
  property: value;
}
```

Note: `a:hover` MUST come after `a:link` and `a:visited` in the CSS definition in order to be effective! `a:active` MUST come after `a:hover` in the CSS definition in order to be effective! Pseudo-class names are not case-sensitive.

All CSS Pseudo Classes

Selector	Example	Example description
:active	a:active	Selects the active link
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the <input> element that has focus
:hover	a:hover	Selects links on mouse over

Activat

background

The `background` shorthand CSS property sets all background style properties at once, such as color, image, origin and size, or repeat method. Component properties not set in the `background` shorthand property value declaration are set to their default values.

Syntax

CSS

```css
/* Using a <background-color> */
background: green;

/* Using a <bg-image> and <repeat-style> */
background: url("test.jpg") repeat-y;

/* Using a <box> and <background-color> */
background: border-box red;

/* A single image, centered and scaled */
background: no-repeat center/80% url("../img/image.png");

/* Global values */
background: inherit;
background: initial;
background: revert;
background: revert-layer;
background: unset;
```

Constituent properties

This property is a shorthand for the following CSS properties:

- `background-attachment`
- `background-clip`
- `background-color`
- `background-image`
- `background-origin`
- `background-position`
- `background-repeat`
- `background-size`

CSS Gradients

CSS gradients let you display smooth transitions between two or more specified colors.

CSS defines three types of gradients:

- **Linear Gradients (goes down/up/left/right/diagonally)**
- **Radial Gradients (defined by their center)**
- **Conic Gradients (rotated around a center point)**

CSS Linear Gradients

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

Syntax

```
background-image: linear-gradient(direction, color-stop1, color-stop2, ...);
```

CSS Radial Gradients

A radial gradient is defined by its center.

To create a radial gradient you must also define at least two color stops.

Syntax

```
background-image: radial-gradient(shape size at position, start-color, ..., last-color);
```

By default, shape is ellipse, size is farthest-corner, and position is center.

Radial Gradient - Evenly Spaced Color Stops (this is default)

CSS Conic Gradients

A conic gradient is a gradient with color transitions rotated around a center point.

To create a conic gradient you must define at least two colors.

Syntax

```
background-image: conic-gradient([from angle] [at position,] color [degree], color [degree], ...);
```

By default, *angle* is 0deg and *position* is center.

If no *degree* is specified, the colors will be spread equally around the center point.


Thank
You!