

## Lesson : Summary

### Introduction to CSS Functions

- CSS functions are used to perform data processing or calculations to return a CSS value for a property.
- These functions represent complex data types and often take input arguments to calculate their return value.

#### attr()

- The attr() function retrieves the value of an attribute from an HTML element and uses it as a property value in CSS.
- Syntax: property: attr(attribute\_name)

#### calc()

- The calc() function performs calculations while defining values for CSS properties.
- Syntax: property: calc(expression)
- Supports standard mathematical operators: addition (+), subtraction (-), multiplication (\*), and division (/).

#### max()

- The max() function sets the largest value from a list of comma-separated expressions as the value of a CSS property.
- Syntax: property: max(value1, value2)

#### min()

- The min() function sets the smallest value from a list of comma-separated expressions as the value of a CSS property.
- Syntax: property: min(value1, value2)

#### var()

- The var() function defines and uses custom CSS variables, allowing for reusable values throughout the stylesheet.
- Syntax:
  - Declare: :root { --variable-name: value; }
  - Use: selector { property: var(--variable-name); }

### 2D Transforms

- In CSS, 2D transformations allow you to manipulate the position, rotation, and scaling of elements in a two-dimensional space on a web page.

#### translate()

- Moves an element along the X and Y axes.
- Syntax: transform: translate(x, y)
- translateX() moves the element along the x-axis.
- translateY() moves the element along the y-axis.

#### scale()

- Changes the width and height of an element.
- Syntax: transform: scale(x, y)
- scaleX() changes the width of an element.
- scaleY() changes the height of an element.

#### rotate()

- Rotates an element based on an angle.

- Syntax: transform: rotate(angle)
- Positive values rotate the element clockwise.
- Negative values rotate the element counterclockwise.

### **skew()**

- Skews the element along the X and Y axes.
- Syntax: transform: skew(x-angle, y-angle)
- skewX() skews the element along the x-axis.
- skewY() skews the element along the y-axis.

### **matrix()**

- Defines a homogeneous 2D transformation matrix.
- Syntax: transform: matrix(a, b, c, d, tx, ty)
- Combines scaling, skewing, and translating in one function.

## **3D Transforms**

3D transformations in CSS allow you to manipulate elements in a three-dimensional space on a web page, adding depth and creating more dynamic visual effects.

### **rotateX()**

- Rotates an element around the X-axis in 3D space.
- Creates a tilting or flipping effect along the horizontal axis.
- Syntax: transform: rotateX(angle)

### **rotateY()**

- Rotates an element around the Y-axis in 3D space.
- Creates a horizontal flipping or spinning effect.
- Syntax: transform: rotateY(angle)

### **rotateZ()**

- Rotates an element around the Z-axis in 3D space.
- Creates a twisting or spinning effect.
- Syntax: transform: rotateZ(angle)

## **What is Transition?**

- CSS transitions enable control over the speed of animation when CSS attributes are modified.
- Components of a CSS transition:
  - Trigger for the animation
  - Start delay
  - Duration of the transition
  - Method of running the transitions

## **Transition Properties**

CSS transitions allow for the smooth change of property values over a specified duration. Key transition properties include:

### **transition-property**

- Specifies which properties will be transitioned.
- Examples:
  - transition-property: none;
  - transition-property: color;
  - transition-property: width;

- transition-property: all;

### transition-duration

- Specifies the duration of the transition in seconds or milliseconds.
- Examples:
  - transition-duration: 6s;
  - transition-duration: 120ms;

### transition-timing-function

- Specifies the timing function used for the transition.
- Predefined timing functions:
  - transition-timing-function: ease;
  - transition-timing-function: ease-in;
  - transition-timing-function: ease-out;
  - transition-timing-function: ease-in-out;

### transition-delay

- Specifies the delay before the transition starts in seconds or milliseconds.
- Examples:
  - transition-delay: 3s;
  - transition-delay: 3000ms;

## Transition Shorthand

- Transitions can be defined in one line using the shorthand syntax.
- Syntax: transition: property\_name duration easing\_function delay

## What is a Filter?

The CSS property called filter is used to apply various graphical effects, such as color shifts or blurs, to an element. Filters are typically employed to modify how images, backgrounds, and borders are displayed.

## Syntax of Filter

To apply filters, use the following syntax:

sql

Copy code

filter: <filter-function>

Filter functions are predefined functions used to apply various visual effects to elements on a web page.

## Common Filter Functions

- blur()
  - Applies a Gaussian blur to the input image.
  - Syntax: filter: blur(radius)
- brightness()
  - Makes the image appear brighter or darker.
  - Syntax: filter: brightness(amount)
- grayscale()
  - Converts the input image to grayscale.
  - Syntax: filter: grayscale(amount)
- hue-rotate()
  - Rotates the hue of an element and its contents.

- Syntax: filter: hue-rotate(angle)
- drop-shadow()
  - Applies a drop shadow effect to the input image or text.
  - Syntax: drop-shadow(offset-x offset-y blur-radius color)

## Combining Multiple Filters

- Multiple filters can be combined to give more visual effects.
- Syntax: filter: <filter-function-1> <filter-function-2>

## What is CSS Animation?

- CSS animations enable transitions between different CSS style configurations.
- Consist of two parts: a style declaration and a collection of keyframes.
- Keyframes define the start, end, and possible intermediate states of the animation.

## Style Declaration

- animation-name: Specifies the name of the animation.
- animation-duration: Defines how long the animation runs (seconds or milliseconds).
- animation-timing-function: Controls animation speed (e.g., linear, ease, ease-in, ease-out).
- animation-delay: Sets a delay before the animation starts.
- animation-iteration-count: Specifies the number of times the animation repeats (e.g., a number or infinite).
- animation-direction: Determines the direction of the animation (normal, reverse, alternate, alternate-reverse).
- animation-fill-mode: Controls animation behavior before and after it runs (forwards, backwards, both, none).
- animation-play-state: Indicates if the animation is running or paused (running, paused).

## Animation Shorthand Syntax

Combines multiple animation properties into one line:

animation: name duration timing-function delay iteration-count direction fill-mode;

•

## Animation Sequence - Keyframes

- Keyframes define the visual changes at different points in the animation sequence.

## Animation vs Transition

Feature	Animation	Transition
Behavior	Moves from initial to final state with intermediate steps.	Moves only from initial to final state.
Iteration	Can set loop count using animation-iteration-count.	Runs only once.
Trigger	Can run automatically or with a trigger.	Runs on a trigger (e.g., hover).

Direction	Runs forward, in reverse, or alternate directions.	Runs forwards on trigger and reverses on trigger release.
Complexity	Ideal for creating complex series of movements.	Suitable for simple movements.

## Introduction to CSS Webkit

- CSS Webkit is an extension specific to Webkit-based browsers like Safari and Google Chrome.
- It provides additional CSS properties prefixed with `-webkit-` to enhance styling capabilities.
- These extensions are designed to work with Webkit's rendering engine to achieve unique visual effects not covered by standard CSS.

## Comparison between Standard CSS and Webkit CSS

- Standard CSS encompasses styles that work across all modern browsers, including Firefox, Edge, and Chrome.
- It adheres strictly to official CSS specifications defined by W3C.
- WebKit CSS, on the other hand, includes properties and values specific to Webkit-based browsers.
- These properties are prefixed with `-webkit-` and are not part of the official CSS standard.
- They are tailored for Safari and Chrome, offering capabilities such as custom scrollbars and specific visual effects.

## Common Webkit Properties

- `-webkit-border-radius`: Adds rounded corners to elements for a modern appearance.
- `-webkit-box-shadow`: Applies drop shadow effects to elements.
- `-webkit-transition`: Creates smooth transitions when CSS properties change.
- `-webkit-transform`: Enables transformations like rotation and scaling.
- `-webkit-overflow-scrolling`: Controls scrolling behavior on touch devices for improved user experience.

## Understanding CSS Custom Properties

- CSS custom properties, also known as CSS variables or cascading variables, are defined by CSS authors to store specific values for reuse throughout a document.
- They are declared using custom property notation, e.g., `--main-color: black;`, and accessed using the `var()` function, e.g., `color: var(--main-color);`.
- Custom properties offer efficiency by centralizing values that are repeated across a stylesheet, simplifying global changes.
- They enhance code readability with semantic identifiers, e.g., `--main-text-color` is more understandable than `#00ff00`.
- Variable names should adhere to CSS naming conventions, starting with double dashes (`--`) and containing only letters and dashes (`-`).