

CSS syllabus

1. Basics of CSS

- What is CSS?
- Adding CSS to HTML (Inline, Internal, External)
- **CSS Selectors**
 - Universal Selector (*)
 - Class Selector (.classname)
 - ID Selector (#idname)
 - Element Selectors
- **CSS Specificity:** Rules for determining which CSS rule takes precedence when multiple selectors apply
 - Inline > id > class > element

2. Colors

- Color Models in CSS is RGB, HEX, color name
- Color, outline-color, caret-color in input tag

3. Background:

- Background Color : background-color: lightblue;
- Background Images: background-image: url('image.jpg');
- Background size: background-size: width px height px;
- Background position: background-position: center;
- Background repeat: background-repeat: no-repeat;
- Background Attachment: background-attachment: fixed;

4. Pseudo-classes and Pseudo-elements

- Pseudo-classes: :hover, :active, :visited etc.
- Pseudo-elements: ::before, ::after

5. Gradients in css

- Linear: background: linear-gradient(to right, red, orange, yellow);
- Radial: background: radial-gradient(circle, red, orange, yellow);
- conic: background: conic-gradient(from 0deg, red, orange, yellow, green, blue);

6. Font and Text Styling

- **Font Properties:**
 - font-family: 'Arial', sans-serif; Sets the font family
 - font-size: 16px; Sets the font size
 - font-weight: bold; Sets the font weight
 - font-style: italic; Sets the font style
- **Text Properties:**
 - color: red; Sets the text color
 - text-align: center; Aligns the text
 - line-height: 1.5; Sets the line height
 - letter-spacing: 2px; Adds space between letters
 - text-decoration: underline; Underlines the text
 - text-transform: uppercase; Converts text to uppercase
 - word-spacing: 5px; Adds space between words

7. Shadow in CSS

- Box Shadow and Text Shadow
 - box-shadow: box-shadow: 5px 5px 5px red, -5px -5px 10px blue;
 - box-shadow: inset 5px 5px 10px red (Creates a shadow inside the element).
 - text-shadow: 2px 2px 5px red;

8. CSS Box Model

- **margin**: Space outside the element's border.
- **border**: The border surrounding the element's padding.
- **padding**: Space inside the element, between the content and the border.
- **Width and height**: Dimensions of the element's content area.

9. Border properties

- border-style, border-style: solid;
- border-width, border-width: 3px;
- border-radius, border-radius: 10px;
- border-color, border-color: red;

10. Layout Techniques

- Display Property: block, inline, inline-block, none
- clip-path: polygon(0 0, 100% 0, 100% 100%, 0% 100%);
- Opacity
 - 0: Fully transparent.
 - 1: Fully opaque.

11. Positioning

- Static,
- Relative
- Absolute
 - Z-Index
- Fixed,
- Sticky

12. Overflow Property

- visible,
- hidden,
- scroll,
- auto

13. CSS Units and Measurements

- Relative Units (em, rem, %)
- Absolute Units (px, cm, mm, etc.)
- Viewport Units (vw, vh)

14. 2D Transformations

a. translate()

- Moves an element from its current position.
- Syntax:
 - transform: translate(x, y)
 - transform : translateX(x)
 - transform: translateY(y)

b. rotate()

- Rotates an element around its origin.
- Syntax: `transform: rotate(angle)`

c. `scale()`

- Scales an element (resizes it).
- Syntax:
 - `transform: scale(x, y)`
 - `transform: scaleX(x)`
 - `transform: scaleY(y)`

d. `skew()`

- Skews an element along the X or Y axis.
- Syntax:
 - `skew(x-angle, y-angle)`
 - `skewX(x-angle)`
 - `skewY(y-angle)`

Note: We can combine multiple transformations by separating them with spaces

Eg. **transform:** `rotate(45deg) translateX(50px) scale(1.5);`

15. Flex box

1. flex-direction

- `flex-direction: row;` */* Default: Items arranged left-to-right */*
- `flex-direction: column;` */* Items arranged top-to-bottom */*
- `flex-direction: row-reverse;` */* Right-to-left */*
- `flex-direction: column-reverse;` */* Bottom-to-top */*

2. justify-content

- `justify-content: flex-start;` */* Default: Items start from the left */*
- `justify-content: flex-end;` */* Items align to the right */*
- `justify-content: center;` */* Items align in the center */*
- `justify-content: space-between;` */* Equal space between items */*
- `justify-content: space-around;` */* Equal space around items */*
- `justify-content: space-evenly;` */* Equal space between and around items */*

3. align-items

- `align-items: flex-start;` */* Items align to the top */*
- `align-items: flex-end;` */* Items align to the bottom */*
- `align-items: center;` */* Items align in the middle */*
- `align-items: stretch;` */* Default: Items stretch to fill container */*
- `align-items: baseline;` */* Align items along their text baselines */*

4. flex-wrap

- `flex-wrap: nowrap;` */* Default: No wrapping */*
- `flex-wrap: wrap;` */* Items wrap to the next line */*
- `flex-wrap: wrap-reverse;` */* Wrapping occurs in reverse */*

5. align-content

- `align-content: flex-start;`
- `align-content: flex-end;`
- `align-content: center;`
- `align-content: space-between;`
- `align-content: space-around;`

6. Order : Specifies the order of the flex items:

- `order:4`

7. Flex Grow : Specifies how much a flex item will grow relative to others, it occupy extra spaces:

- flex-grow: 1;
8. flex-shrink : It determines how much a flex item **shrinks** when there isn't enough space in the flex container.
- flex-shrink: 2; /* Shrinks faster */ flex-shrink: 1; /* Shrinks normally */

16. Transition for Animation effects:

- Transition-property:
 - The CSS property to transition (e.g., color, background-color, transform).
 - Eg transition-property: background-color;
 - transition-property: all;
- Transition-duration:
 - The time that transition takes (e.g., 2s, 500ms)
 - Eg. transition-duration: 2s;
- Transition-timing-function:
 - Controls the speed of the transition effect
 - ease: (Default) Start and end slow, fast in middle.
 - linear: Constant speed (all transition at same speed)
 - ease-in: Start slow, then go fast
 - ease-out: Start fast, then go slow
 - ease-in-out: Start and end slow, fast in middle
 - cubic-bezier: It is use to define own speed
 - Eg. **transition-timing-function**: ease-in-out
- Transition-delay:
 - Time before the transition starts
 - transition-delay: 1s

Note: We can combine multiple transition by separating them with spaces

transition: [property] [duration] [timing-function] [delay];

Eg. transition: background-color 2s ease-in 1s;

17. Keyframes for Animations effects:

- animation-name:
 - It define the name of animation.
 - animation-name: slideIn
- animation-duration:
 - The time that animation takes (e.g., 2s, 500ms)
 - Eg. animation-duration: 3s
- animation-timing-function
 - Controls the speed of the animation effect. It is same as transition-timing-function
 - Eg. animation-timing-function: linear;
- animation-delay
 - Time before the animation starts
 - Eg. animation-delay: 2s;
- animation-iteration-count
 - It defines total no of animation count it can also be infinite or numerical value
 - Eg. animation-iteration-count: infinite or 3.
- animation-direction.
 - It define the direction of animation, normal, reverse and alternate

- Eg animation-direction: alternate;

Note: We can combine multiple animation by separating them with spaces

animation: [animation-name] [duration] [timing-function] [delay] [iteration-count] [direction];

Eg. animation: slideIn 2s linear 0s infinite alternate;

Keyframe: keyframes defines the animation's points (percentages) and states (CSS properties) of animation duration. It means which CSS properties will show at each percentage points in animation duration.

```
@keyframes animationName {
  0%{

  }
  25%{

  }
  50%{

  }
  75%{

  }
  100%{

  }
}
```

18. CSS Variables and Custom Properties

- Defining and Using Variables
 - --variablename: value;
 - * {

--MyColor: red
 --MyFontSize: 16px;
- The **var()** Function: It is use to execute variable's value in style properties
 - .parent{

font-size: var(MyFontSize);
 background-color: var(--MyColor);

19. Responsive Design

- min-width, max-width.
- Media Queries

@media (max-width: 500px) { body { background-color: lightblue; } }	@media (min-width: 500px) and (max-width: 900px) { body { background-color: lightgreen; } }	@media (min-width: 900px) { body { background-color: yellow; } }
---	---	--

20. Grid Layout

- Introduction to CSS Grid.
- Grid container, grid items, grid lines.

1. Grid container

- `display: grid;`
- `grid-template-columns: repeat(3,1fr)` : Defines the number of columns.
- `grid-template-rows: 200px 200px`
- `grid-template-columns: repeat(auto-fill, 250px)` : For responsive design
- `grid-gap`, `row-gap` and `column-gap`

2. Grid Item Placement

- `grid-column-start:1`
- `grid-column-end:2`
- `grid-row-start`
- `grid-row-end`
- `grid-column:1/3` :Shorthand for grid placement
- `grid-row:1/3`
- `grid-area: 1/1/1/1` :Start row/start column/end row/end column

3. justify-content (use for Horizontal)

- `justify-content: start;` /* Default: Items start from the left */
- `justify-content: end;` /* Items align to the right */
- `justify-content: center;` /* Items align in the center */
- `justify-content: space-between;` /* Equal space between items */
- `justify-content: space-around;` /* Equal space around items */
- `justify-content: space-evenly;` /* Equal space between and around items */

4. align-items (use for vertical)

- `align-items: start;` /* Items align to the top of grid box */
- `align-items: end;` /* Items align to the bottom of grid box*/
- `align-items: center;` /* Items align in the middle of grid box */

5. align-content (use for Vertical)

- `align-content: start;` /* Items align to the top of parent box*/
- `align-content: end;` /* Items align to the bottom of parent box*/
- `align-content: center;` /* Items align to the center of parent box*/
- `align-content: space-between;` /* Equal space between items in vertical*/
- `align-content: space-around;` /* Equal space around items in vertical*/
- `align-content: space-evenly;` /* Equal space between and around items */