

```
/*
 * CSS - COMPLETE MASTERY GUIDE (10-PAGE FORMAT)
 * GOAL: 100% knowledge, in-depth details, proper organization
 */
```

```
/*
```



---

---

PAGE 1  
CSS FUNDAMENTALS

---

---



```
*/
//--- WHAT IS CSS? ---
// CSS (Cascading Style Sheets) ek style sheet language hai jo HTML elements ko
style karta hai.
// CSS decides how HTML elements display on screen, paper, or other media.
// CSS saves a lot of work by controlling multiple pages' layout at once.
```

```
//--- HOW TO ADD CSS ---
// 1. External CSS - Separate .css file (recommended):
<link rel="stylesheet" href="styles.css"> // HTML <head> me add karein
```

```
// 2. Internal CSS - HTML file ke <head> section me:
```

```
<style>
  body { background-color: lightblue; }
  p { color: red; }
</style>
```

```
// 3. Inline CSS - Directly in HTML elements:
```

```
<p style="color: red; margin-left: 20px;">This is a paragraph.</p>
```

```
//--- CSS SELECTORS ---
```

```
// Element Selector:
```

```
p { color: red; } // Saare <p> elements ko target karta hai
```

```
// ID Selector (#):
```

```
#header { background-color: blue; } // <div id="header"> ko target karta hai
```

```
// Class Selector (.):
```

```
.intro { font-size: 18px; } // <p class="intro"> jaisi elements ko target karta
hai
```

```
// Universal Selector (*):
```

```
* { margin: 0; padding: 0; } // Saare elements ko target karta hai
```

```
// Grouping Selector:
```

```
h1, h2, p { text-align: center; } // Multiple elements ko ek saath target karta
hai
```

```
//--- CSS COMMENTS ---
```

```
/* This is a multi-line comment */
```

```
// This is a single-line comment (not standard CSS, but works in some
preprocessors)
```

```
/*
```



---

---

PAGE 2  
CSS PROPERTIES: TEXT

---

---



```
*/
//--- TEXT PROPERTIES ---
// Color:
color: red; // Named color
color: #ff0000; // Hex color
color: rgb(255, 0, 0); // RGB color
```

```

color: rgba(255, 0, 0, 0.5);    // RGB with alpha (transparency)
color: hsl(0, 100%, 50%);      // HSL color (hue, saturation, lightness)
color: hsla(0, 100%, 50%, 0.5); // HSL with alpha

// Text Alignment:
text-align: left;                // Default
text-align: right;
text-align: center;
text-align: justify;            // Full alignment (both edges)

// Text Decoration:
text-decoration: none;          // No decoration
text-decoration: underline;
text-decoration: overline;
text-decoration: line-through;
text-decoration: underline red wavy; // Combined (line, color, style)

// Text Transform:
text-transform: none;           // No transformation
text-transform: uppercase;      // ALL UPPERCASE
text-transform: lowercase;      // all lowercase
text-transform: capitalize;     // First Letter Uppercase

// Text Spacing:
text-indent: 50px;              // First line indentation
letter-spacing: 3px;            // Space between characters
line-height: 1.8;               // Space between lines
word-spacing: 10px;             // Space between words
white-space: nowrap;            // Prevents text wrapping

// Text Shadow:
text-shadow: 2px 2px 5px red;   // horizontal, vertical, blur, color

// Direction:
direction: ltr;                 // Left to right (default)
direction: rtl;                 // Right to left

/*

```



```

*/
//--- FONT PROPERTIES ---
// Font Family:
font-family: "Times New Roman", Times, serif; // Multiple options (fallbacks)
font-family: Arial, Helvetica, sans-serif;
font-family: "Courier New", Courier, monospace;
// Generic families: serif, sans-serif, monospace, cursive, fantasy, system-ui

// Font Size:
font-size: 16px;                // Absolute size in pixels
font-size: 1em;                 // Relative to parent (1em = parent's font size)
font-size: 1rem;                // Relative to root element (html)
font-size: 80%;                 // Percentage of parent's font size
font-size: large;               // Named size
// Relative sizes: xx-small, x-small, small, medium, large, x-large, xx-large
// Relative sizing: smaller, larger (compared to parent element)

// Font Weight:
font-weight: normal;             // Normal thickness (400)
font-weight: bold;              // Bold thickness (700)
font-weight: 100;               // Thin

```

```

font-weight: 900; // Extra bold
// Values: 100, 200, 300, 400 (normal), 500, 600, 700 (bold), 800, 900

// Font Style:
font-style: normal; // Normal
font-style: italic; // Italic
font-style: oblique; // Oblique (similar to italic but less supported)

// Font Variant:
font-variant: normal;
font-variant: small-caps; // Small capitals

//--- WEB FONTS ---
// Google Fonts (HTML):
<link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">

// Google Fonts (CSS):
@import url('https://fonts.googleapis.com/css?family=Open+Sans');

// Custom Web Fonts:
@font-face {
    font-family: "MyFont";
    src: url("myfont.woff2") format("woff2"),
        url("myfont.woff") format("woff");
    font-weight: normal;
    font-style: normal;
}

// Font shorthand:
font: italic bold 16px/1.8 Arial, sans-serif; // style weight size/line-height family

```

/\*



```

*/
//--- BOX MODEL ---
// Every HTML element is a rectangular box with:
// Content -> Padding -> Border -> Margin

// Width & Height:
width: 300px; // Content width
height: 200px; // Content height
min-width: 100px; // Minimum width
max-width: 600px; // Maximum width
min-height: 100px; // Minimum height
max-height: 600px; // Maximum height

// Margin (Outside of border):
margin: 10px; // All sides 10px
margin: 10px 20px; // Top/bottom 10px, left/right 20px
margin: 10px 20px 30px; // Top 10px, left/right 20px, bottom 30px
margin: 10px 20px 30px 40px; // Top, right, bottom, left (clockwise)
margin-top: 10px; // Individual side
margin-right: 20px;
margin-bottom: 30px;
margin-left: 40px;
margin: auto; // Center horizontally (with width)
margin: 0 auto; // Center horizontally with top/bottom 0

// Padding (Between content and border):

```

```
padding: 10px;                // All sides 10px
padding: 10px 20px;           // Top/bottom 10px, left/right 20px
padding: 10px 20px 30px;      // Top 10px, left/right 20px, bottom 30px
padding: 10px 20px 30px 40px; // Top, right, bottom, left
padding-top: 10px;            // Individual side
padding-right: 20px;
padding-bottom: 30px;
padding-left: 40px;

// Border:
border: 1px solid black;      // Width, style, color (all sides)
border-width: 1px;
border-style: solid;          // solid, dotted, dashed, double, groove,
                                // ridge, inset, outset
border-color: black;
border-top: 1px solid black;   // Individual side
border-right: 2px dashed red;
border-bottom: 3px dotted green;
border-left: 4px double blue;
border-radius: 10px;          // Rounded corners (all corners)
border-top-left-radius: 5px;  // Individual corner

// Box Sizing:
box-sizing: content-box;      // Default (width/height = content only)
box-sizing: border-box;       // Width/height includes padding and border
```

/\*



```
*/
//--- DISPLAY PROPERTY ---
display: block;                // Full width, line break before & after
display: inline;               // Only takes needed width, no line breaks
display: inline-block;         // Like inline but can set width/height
display: none;                 // Removes from document flow (not visible)
display: flex;                 // Flexible layout (parent becomes flex
                                // container)
display: grid;                 // Grid layout
display: table;                // Behaves like <table>
display: table-cell;           // Behaves like <td>

//--- VISIBILITY PROPERTY ---
visibility: visible;           // Default
visibility: hidden;            // Not visible but takes up space (unlike
display:none)
visibility: collapse;          // For table elements

//--- POSITION PROPERTY ---
position: static;              // Default (follows document flow)
position: relative;            // Positioned relative to normal position
position: absolute;            // Positioned relative to nearest positioned
                                // ancestor
position: fixed;               // Positioned relative to viewport (stays while
                                // scrolling)
position: sticky;              // Positioned based on scroll position (hybrid
                                // of relative/fixed)

// Position Coordinates:
top: 10px;                    // Distance from top
right: 20px;                  // Distance from right
bottom: 30px;                 // Distance from bottom
left: 40px;                   // Distance from left
```

```

z-index: 1; // Stacking order (higher number = in front)

//--- FLOAT & CLEAR ---
float: left; // Element floats to the left
float: right; // Element floats to the right
float: none; // Default (no floating)

clear: left; // No floating elements allowed on left side
clear: right; // No floating elements allowed on right side
clear: both; // No floating elements allowed on either side
clear: none; // Default

//--- OVERFLOW ---
overflow: visible; // Content overflows the box (default)
overflow: hidden; // Overflowing content is clipped
overflow: scroll; // Always shows scrollbars
overflow: auto; // Shows scrollbars only when needed
overflow-x: hidden; // Controls horizontal overflow
overflow-y: scroll; // Controls vertical overflow

```

/\*



```

*/
//--- COLORS ---
// Ways to specify colors:
color: red; // Named color
color: #ff0000; // Hex color
color: rgb(255, 0, 0); // RGB color
color: rgba(255, 0, 0, 0.5); // RGB with alpha (0.0 - 1.0)
color: hsl(0, 100%, 50%); // HSL (hue: 0-360, saturation/lightness: 0-100%)
color: hsla(0, 100%, 50%, 0.5); // HSL with alpha

// Color keywords:
transparent; // Fully transparent
currentcolor; // Current color value
inherit; // Inherit from parent

//--- BACKGROUNDS ---
// Background Color:
background-color: lightblue; // Single color background

// Background Image:
background-image: url("image.jpg");
background-image: linear-gradient(to right, red, yellow); // Gradient
background-image: radial-gradient(circle, red, yellow, green); // Radial gradient

// Multiple Background Images:
background-image: url("img1.jpg"), url("img2.jpg"); // First listed is on top

// Background Repeat:
background-repeat: repeat; // Default (repeats both x and y)
background-repeat: repeat-x; // Repeats horizontally
background-repeat: repeat-y; // Repeats vertically
background-repeat: no-repeat; // No repetition

// Background Position:
background-position: top left; // Named positions
background-position: center center;
background-position: 50% 50%; // Percentage

```

```
background-position: 10px 20px; // Pixel coordinates (from top-left)

// Background Size:
background-size: auto;           // Default (original size)
background-size: cover;          // Cover entire area (may crop)
background-size: contain;        // Fit inside area (may leave gaps)
background-size: 100% 100%;      // Stretch to fit
background-size: 200px 100px;    // Specific dimensions

// Background Attachment:
background-attachment: scroll;    // Default (scrolls with page)
background-attachment: fixed;    // Fixed in viewport
background-attachment: local;    // Scrolls with element's contents

// Background Shorthand:
background: lightblue url("img.jpg") no-repeat center top / cover fixed;
// color image repeat position/size attachment
```

```
/*
```



```
*/
//--- FLEXBOX ---
// Flexbox is a one-dimensional layout method for arranging items.
// Container Properties:
display: flex;                // Makes an element a flex container
flex-direction: row;           // Default (left to right)
flex-direction: row-reverse;   // Right to left
flex-direction: column;        // Top to bottom
flex-direction: column-reverse; // Bottom to top

flex-wrap: nowrap;             // Default (single line)
flex-wrap: wrap;               // Multiple lines if needed
flex-wrap: wrap-reverse;       // Multiple lines, opposite direction

flex-flow: row wrap;           // Shorthand for flex-direction + flex-wrap

justify-content: flex-start;    // Default (items at start)
justify-content: flex-end;      // Items at end
justify-content: center;        // Items at 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

align-items: stretch;          // Default (stretch to fill container height)
align-items: flex-start;        // Items at top
align-items: flex-end;          // Items at bottom
align-items: center;            // Items at vertical center
align-items: baseline;          // Items aligned by their baselines

align-content: flex-start;      // Lines at top (for multi-line flex)
align-content: flex-end;        // Lines at bottom
align-content: center;          // Lines at center
align-content: space-between;   // Equal space between lines
align-content: space-around;    // Equal space around lines
align-content: stretch;        // Default (lines stretch to fill)

// Flex Item Properties:
order: 0;                       // Default (source order)
order: 1;                       // Appears later in order
order: -1;                      // Appears earlier in order
```

```

flex-grow: 0;           // Default (doesn't grow)
flex-grow: 1;           // Grows to fill available space
flex-grow: 2;           // Grows twice as much as flex-grow:1

flex-shrink: 1;         // Default (can shrink)
flex-shrink: 0;         // Doesn't shrink

flex-basis: auto;       // Default (based on content)
flex-basis: 200px;      // Initial main size

flex: 0 1 auto;         // Shorthand (grow shrink basis)
flex: 1;                // Same as flex: 1 1 0%

align-self: auto;       // Default (follows container's align-items)
align-self: flex-start; // Overrides for individual item
align-self: flex-end;
align-self: center;
align-self: baseline;
align-self: stretch;

//--- CSS GRID ---
// Grid is a two-dimensional layout system.
// Grid Container Properties:
display: grid;           // Makes an element a grid container
grid-template-columns: 100px 200px 100px; // Three columns with sizes
grid-template-columns: 1fr 2fr 1fr; // Fraction units
grid-template-columns: repeat(3, 1fr); // Three equal columns
grid-template-columns: auto 100px 1fr; // Mixed units

grid-template-rows: 100px 200px; // Two rows with sizes
grid-template-rows: repeat(3, 100px); // Three equal rows

grid-template-areas:      // Named grid areas
    "header header header"
    "sidebar content content"
    "footer footer footer";

grid-column-gap: 10px;    // Space between columns
grid-row-gap: 15px;       // Space between rows
grid-gap: 15px 10px;      // Shorthand (row-gap column-gap)
gap: 15px 10px;           // Modern shorthand

justify-items: start;     // Horizontal alignment within cells
justify-items: end;
justify-items: center;
justify-items: stretch;  // Default

align-items: start;        // Vertical alignment within cells
align-items: end;
align-items: center;
align-items: stretch;     // Default

// Grid Item Properties:
grid-column-start: 1;      // Start at first grid line
grid-column-end: 3;        // End at third grid line
grid-column: 1 / 3;        // Shorthand (start / end)
grid-column: 1 / span 2;   // Start at 1, span 2 columns

grid-row-start: 1;
grid-row-end: 3;
grid-row: 1 / 3;           // Shorthand (start / end)

grid-area: header;         // Corresponds to template area name
grid-area: 1 / 1 / 3 / 3;  // Shorthand (row-start/col-start/row-end/col-

```

```

end)

justify-self: start;           // Horizontal alignment (individual item)
justify-self: end;
justify-self: center;
justify-self: stretch;       // Default

align-self: start;           // Vertical alignment (individual item)
align-self: end;
align-self: center;
align-self: stretch;       // Default

```

```
/*
```



```

*/
//--- MEDIA QUERIES ---
// Media queries allow content to adapt to different devices/screen sizes.
@media screen and (max-width: 600px) {
    body {
        background-color: lightblue;
    }
}

// Media Types:
@media screen { }           // Computer screens
@media print { }           // Print preview & printed pages
@media speech { }          // Screen readers
@media all { }             // All media types (default)

// Common Features:
@media (max-width: 600px) { } // Width <= 600px
@media (min-width: 600px) { } // Width >= 600px
@media (min-width: 600px) and (max-width: 900px) { } // 600px <= Width <= 900px
@media (orientation: landscape) { } // Width > Height
@media (orientation: portrait) { } // Height >= Width
@media (hover: hover) { }    // Device capable of hovering

// Common Breakpoints:
@media (max-width: 600px) { } // Small devices (phones)
@media (min-width: 600px) { } // Medium devices (tablets)
@media (min-width: 992px) { } // Large devices (desktops)
@media (min-width: 1200px) { } // Extra large devices

//--- RESPONSIVE IMAGES ---
img {
    max-width: 100%;           // Image won't be larger than container
    height: auto;             // Preserve aspect ratio
}

// Picture element (HTML):
/*
<picture>
  <source media="(max-width: 600px)" srcset="small.jpg">
  <source media="(max-width: 1200px)" srcset="medium.jpg">
  
</picture>
*/

//--- RESPONSIVE TYPOGRAPHY ---
// Using relative units:
body {

```



```

    font-size: 16px;           // Base size
}

h1 {
    font-size: 2em;           // 2x the parent element's font size
}

// Using viewport units:
h1 {
    font-size: 5vw;           // 5% of viewport width
}

// Fluid typography with calc():
h1 {
    font-size: calc(16px + 2vw); // Minimum 16px, grows with viewport
}

//--- VIEWPORT META TAG (HTML) ---
/*
<meta name="viewport" content="width=device-width, initial-scale=1.0">
*/

/*

```



```

*/
//--- TRANSFORMS ---
// 2D Transforms:
transform: translate(50px, 100px); // Move right 50px, down 100px
transform: translateX(50px);       // Move horizontally
transform: translateY(100px);      // Move vertically

transform: scale(2);               // Scale 2x in both dimensions
transform: scaleX(2);              // Scale horizontally
transform: scaleY(0.5);            // Scale vertically

transform: rotate(45deg);          // Rotate 45 degrees clockwise

transform: skew(20deg, 10deg);     // Skew along X and Y axes
transform: skewX(20deg);           // Skew horizontally
transform: skewY(10deg);           // Skew vertically

// Multiple Transforms:
transform: translate(50px, 50px) rotate(45deg) scale(1.5);

// 3D Transforms:
transform: rotateX(45deg);         // Rotate around X-axis
transform: rotateY(45deg);         // Rotate around Y-axis
transform: rotateZ(45deg);         // Rotate around Z-axis (same as
rotate(45deg))

transform: perspective(100px);     // 3D perspective

// Transform Origin:
transform-origin: center;          // Default
transform-origin: top left;        // Named position
transform-origin: 50% 50%;         // Percentage
transform-origin: 10px 20px;       // Pixels

//--- TRANSITIONS ---
// Transitions allow property changes to occur smoothly over time:
transition: property duration timing-function delay;

```

```

transition-property: all;           // Which properties to animate (default: all)
transition-property: width, height; // Specific properties
transition-duration: 2s;           // How long the transition takes
transition-timing-function: ease;  // Speed curve (ease, linear, ease-in, ease-
out, ease-in-out)
transition-delay: 1s;              // Wait time before starting

// Shorthand:
transition: width 2s ease-in-out, height 2s ease 1s;

//--- ANIMATIONS ---
// Define animation:
@keyframes slide {
  0% {                             // Start state (can use "from" instead)
    transform: translateX(0);
  }
  50% {                             // Middle state
    transform: translateX(50px);
  }
  100% {                             // End state (can use "to" instead)
    transform: translateX(100px);
  }
}

// Apply animation:
animation-name: slide;              // Name of the @keyframes
animation-duration: 3s;             // How long the animation takes
animation-timing-function: ease;    // Speed curve
animation-delay: 1s;               // Wait time before starting
animation-iteration-count: 3;       // How many times (infinite for looping)
animation-direction: alternate;    // Direction (normal, reverse, alternate,
alternate-reverse)
animation-fill-mode: forwards;     // What values apply when animation isn't
running                             // (none, forwards, backwards, both)
animation-play-state: running;     // Whether animation is playing or paused

// Shorthand:
animation: slide 3s ease 1s infinite alternate forwards;

```

/\*



```

*/
//--- ADVANCED SELECTORS ---
// Descendant Selector (space):
div p { color: red; }              // All p inside any div

// Child Selector (>):
div > p { color: red; }            // p that are direct children of div

// Adjacent Sibling Selector (+):
h1 + p { color: red; }             // p immediately after h1

// General Sibling Selector (~):
h1 ~ p { color: red; }             // All p that follow h1 (same parent)

// Attribute Selectors:
[href] { color: blue; }            // Elements with href attribute
[href="https://example.com"] { color: blue; } // Exact match
[href^="https"] { color: green; }  // Starts with

```

```

[href$=".pdf"] { color: red; } // Ends with
[href*="example"] { color: purple; } // Contains
[href~="sample"] { color: orange; } // Contains word
[lang]="en"] { color: cyan; } // Starts with "en" or "en-"

// Pseudo-classes (element states):
a:link { color: blue; } // Unvisited links
a:visited { color: purple; } // Visited links
a:hover { color: red; } // Mouse over
a:active { color: green; } // Being clicked

input:focus { border: 2px solid blue; } // When element has focus
input:checked { margin-left: 25px; } // Checked inputs
input:disabled { background: #ccc; } // Disabled inputs
div:first-child { font-weight: bold; } // First child of its parent

// Pseudo-elements (parts of elements):
p::first-line { font-weight: bold; } // First line of p
p::first-letter { font-size: 200%; } // First letter of p
p::before { content: "Read: "; } // Insert before content
p::after { content: " [more]"; } // Insert after content
::selection { background: yellow; } // Selected text

//--- CSS VARIABLES (CUSTOM PROPERTIES) ---
:root {
  --main-color: #4CAF50; // Define variable at root level
  --main-padding: 15px;
}

div {
  background-color: var(--main-color); // Use variable
  padding: var(--main-padding);
  border: 1px solid var(--main-color, black); // With fallback
}

//--- SPECIFICITY & CASCADE ---
// Priority order (lowest to highest):
// 1. Browser defaults
// 2. External & internal stylesheets (cascade order)
// 3. Inline styles
// 4. !important declarations

// Specificity calculation:
// - Inline styles: 1000 points
// - ID selectors: 100 points
// - Class selectors, attributes, pseudo-classes: 10 points
// - Element selectors, pseudo-elements: 1 point

// Example:
#nav .selected p { } // 1 ID + 1 class + 1 element = 111 points
.nav p.selected { } // 2 classes + 1 element = 21 points

// !important (avoid if possible):
p { color: red !important; } // Overrides other declarations

//--- BEST PRACTICES ---
// 1. Use CSS reset or normalize.css
// 2. Organize CSS with comments and logical grouping
// 3. Use meaningful class names (BEM, OOCSS, etc.)
// 4. Avoid excessive nesting
// 5. Use shorthand properties when appropriate
// 6. Minimize use of !important
// 7. Use relative units (em, rem, %, vw/vh) for flexibility
// 8. Consider browser compatibility (can use prefixes or tools like

```

```
Autoprefixer)
// 9. Test on multiple devices and browsers
// 10. Minify CSS for production
```

```

/*****
*
*                               END OF MASTERY GUIDE                               *
*****/
```