Basic Syntax: The structure of a CSS rule consists of a selector and a declaration block.

```
selector {
    property: value;
}
```

Example:

```
<!DOCTYPE html>
<html>
<head>
    <style>
        p{
            color: red;
        }
    </style>
</head>
<body>
    <p>hell</p>
</body>
</html>
```

Selectors: Selectors are used to target HTML elements you want to style.

Selector	Meaning	Example
*	All elements	* { margin: 0; }
elements	Specific	P {}
#id	ID selector	#header {}
.class	Class selector	.box {}
element1, element2	Multiple elements	h1l, h2 {}
element1 element2	Descendant selector	div p {}
element > element	Direct child	ul > li {}
element + element	Adjacent sibling	h1 + p {}

Properties: Properties define how HTML elements are displayed.

Text: These properties control font, color, alignment, and text decoration.

Property	Description	Example
color	Text color	color: blue;
font-size	Size of text	font-size: 16px;
font-family	Typeface	font-family: Arial, sans-serif;
text-align	Text alignment	text-align: center;
text-decoration	Underline, overline, none	text-decoration: underline;
font-weight	Boldness (normal, bold, etc.)	font-weight: bold;
line-height	Space between lines	line-height: 1.5;

Box Model: The box model describes the structure of elements including content, padding, border, and margin.

Property	Description	Example
width	Width of element	width: 200px;
height	Height of element	height: 100px;
padding	Space inside border	padding: 10px;
margin	Space outside border	margin: 20px;
border	Border style	border: 1px solid red;
box-sizing	Include padding in size	box-sizing: border-box;

Colors & Backgrounds: These properties define element background, image, color, and opacity.

Property	Description	Example
background-color	Element background color	background-color: #000000
background-image	Background image	background-image: url("img.jpg");
background-size	Size of background image	background-size: cover;
color	Text color	color: #333;
opacity:	Transparency (0 to 1)	opacity: 0.5;

Positioning: Positioning controls how elements are placed on the page.

Property	Description	Example
position	static, relative, absolute, and fixed	position: absolute;
top, left, right, bottom	Offset values	top: 10px;
z-index	Stack order	z-index: 100;

Display & Flexbox: These properties define how elements are displayed and laid out in a container.

Property	Description	Example
display	Layout type	display: block; or display: flex;
flex-direction	Row or column layout	flex-direction: center;
justify-content	Horizontal alignment	justify-content: center;
align-items	Vertical alignment	align-items: center;
gap	Space between flex items	gap: 10px;

```html
<!DOCTYPE html>
<html>
<head>
    <style>
        .container{
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: darksalmon;
            gap: 20px;
        }
        .box{
            width: 100px;
            height: 100px;
            background-color: coral;
            display: flex;
            justify-content: center;
            align-items: center;
            font-weight: bold;
            color: white;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="box">Box 1</div>
        <div class="box">Box 2</div>
        <div class="box">Box 3</div>
    </div>
</body>
</html>
```

Grid: Grid is a layout system for creating rows & columns

Property	Description	Example
display: grid	Grid container	display: grid;
grid-template-columns	Define column layout	grid-template-columns: 1fr 2fr;
gap	Space between grid items	gap: 20px;

```html
<!DOCTYPE html>
<html>
<head>
    <style>
        .grid-container{
            display: grid;
            grid-template-columns: repeat(3, 1fr);  /* 3 equal columns */
            gap: 1px;
            padding: 10px;
        }
        .grid-item{
            background-color: ☐black;
            color: ■white;
            padding: 20px;
            text-align: center;
            border-radius: 10px;
        }
    </style>
</head>
<body>
    <div class="grid-container">
        <div class="grid-item">1</div>
        <div class="grid-item">2</div>
        <div class="grid-item">3</div>
        <div class="grid-item">4</div>
        <div class="grid-item">5</div>
        <div class="grid-item">6</div>
    </div>
</body>
</html>
```

Transitions & Animations: These properties allow smooth changes and animations of element styles.

Property	Description	Example
transition	Smooth property change	transition: all 0.3s ease;
@keyframes	Animation sequence	See below

```html
<!DOCTYPE html>
<html>
<head>
    <style>
        .box{
            width: 100px;
            height: 100px;
            background-color: ■coral;
            transition: all 5s easy;
            justify-items: center;
            align-items: center;
        }
        .box:hover{
            background-color: ■lightblue;
            transform: scale(5);
        }
    </style>
</head>
<body>
    <!-- transform example -->
    <div class="box"></div>
</body>
</html>
```

```html
<!DOCTYPE html>
<html>
<head>
    <style>
        @keyframes fadeIn {
            from {opacity: 0;}
            to {opacity: 1;}
        }
        .box{
            width: 100px;
            height: 100px;
            background-color: ■mediumseagreen;
            animation: fadeIn 10s ease-in-out;
        }
    </style>
</head>
<body>
    <!-- animation example -->
    <div class="box"></div>
</body>
</html>
```

Pseudo-classes & Elements: Pseudo-classes target elements based on their state; pseudo-elements target parts of elements.

Selector	Description	Example
:hover	When mouse hovers	a:hover {color: red;}
:first-child	First child element	li:first-child {}
:nth-child(n)	nth child (1-based)	li:nth-child(2)
::before	Insert content before element	h1::before {content: "* ";}
::after	Insert content after element	h1::after {content: ".";}

Media Queries (Responsive Design): Media queries apply styles depending on screen size and device type.

```html
<!DOCTYPE html>
<html>
<head>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: ■white;
            color: ■black;
            text-align: center;
            padding: 50px;
        }

        h1 {
            font-size: 36px;
        }

        /* Media Query: For screens 600px wide or less */
        @media (max-width: 600px) {
            body {
                background-color: ■#add8e6; /* Light Blue */
                color: ■darkblue;
            }

            h1 {
                font-size: 24px;
            }
        }
    </style>
</head>
<body>
    <h1>Hello, Responsive World!</h1>
    <p>Resize the browser window to see the style change.</p>
</body>
</html>
```