CSS Short Hand

Types	Example
Inline css	<h1 style="color:green;">A Green Heading</h1>
Internal css	<style> body {background-color: pink;} </style>

External css	<link rel="stylesheet" href="style.css" type=”text/css”>
	
	In style.css
	body
	{
	background-color: pink;
	}

Selectors

Type	Example
Element	p{…}, img{…}, div{…} etc
Id	#idname{…}
Class	.classname{…}
Universal	*{…}
Descendant	ul b{…}
Child	ul > b{…}
Attribute	input [type=”text”]{….}

Color

Method	Values	Example
Keyword	Name of color	<h2 style="color:orange;">
Hex code	#rrggbb	<h5 style="color: #df0fdd;">
RGB	rgb()	<h4 style="color:rgb(25,140,230);">

Background

Method	Values	Example
Background-color		pre {background-color: aqua}

Background-image	URL	body { background-image: url("scenary.jfif");
		}

Background-repeat
By default, the background-image property repeats an image both horizontally and vertically.
	repeat-x,
	repeat-y,
	no-repeat
		body { background-image: url("nature.jfif");
		background-repeat: repeat-X;
		}

shorthand		background: url(“scenary.jfif”) pink repeat-x

CSS fonts

Property-name	Values	Example

Font-family	Times New Roman, Arial, Verdana, etc	.p2 {
		font-family:Arial, Helvetica, sans-serif;
		}

Font style	normal, italic	p.italic {font-style: italic;}
Font-weight	normal, bold	.thick { font-weight: bold;}
Font-variant	normal, small-caps (all lowercase letters are converted to uppercase letters)	p.small { font-variant: small-caps;}
Font-size	medium|xx-small|x-small|small|large|x-large|xx-large|px|%	.c { font-size: 150%;}

CSS text manipulation

Property-name	Values	Example

text-indent
specifies the indentation of the first line in a text-block.
	px,%
	Negative values are allowed.
		.a {
		text-indent: 50px;
		}

text-decoration	none, underline, overline, line-through	h1 {
		text-decoration: overline;
		}

text-transform	None, capitalize *(first character of each word to uppercase)*, lowercase, uppercase *(all characters to uppercase)*	.a {
		text-transform: uppercase;
		}

text-align	Left, right, center, justify	.a {
		text-align: center;
		}

letter-spacing	px,
	Negative value allowed
		h2 {
		letter-spacing: 5px;
		}

word-spacing	px,
	Negative value allowed
		h2{
		word-spacing: 10px;
		}

CSS Borders:

Property-name	Values	Example
Border-style	solid, dotted, dashed, double	p.dotted {border-style: dotted;} p.mix {border-style: dotted dashed solid double;}
Border-color		p.three {border-style: dotted; border-color: red green;}
Border-width	Px,thick, medium, thin	p.three { border-style: dotted; border-width: 2px;}
Short-hand		.a {border: 5px solid red;}

Pseudo classes:

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

Class Type	Example
:link {}	a:link {color: red;}
:visited{}	a:visited {color: green;}
:hover{}	a:hover {color: hotpink;}
:active{}	a:active {color: blue;}

Note: a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.

Note: a:active MUST come after a:hover in the CSS definition in order to be effective.

Pseudo elements:

Pseudo-element	Values	Example
::first-line		p::first-line{font-size:30px;font-style:italic;font-weight:bold;}
::first-letter		p::first-letter{font-size:100px;}
::before	URL	h1::before{content:url("smile.png");}
::after	URL	h1::after{content:”Hello”);}
::marker (selects the markers of list items.)		::marker{color:red; font-size:20px;}

::selection Pseudo-element	color, background and outline.	::selection
		{
		color:green;
		background-color:yellow;
		}

CSS Box model:

The box model allows us to add a border around elements, and to define space between elements. The CSS box model is essentially a box that wraps around every HTML element. It consists of: margin, border, padding, and the actual content.

Method	Values	Example
Margin (Clears an area outside the border)	Px	<p style="margin:10px; padding:50px 50px; border:5px dashed red"> THIS IS P2 </p>
Padding (Clears an area around the content)	px	

Side wise margin/padding can be given both ways

1. Margin/padding : 10px 20px 30px 20 px (clockwise rotation : Top 10px / right 20 px / bottom 30px / left 40px)
2. Margin-top/margin-bottom/margin-right/margin-left (Same for padding*)

CSS positioning:

Method	Values	Example

Absolute (is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed))		.p3{
		border:2px solid purple;
		position:fixed; top:20px;
		left:30px;}

Relative		
Fixed (is positioned relative to the viewport)		
Static	By default	

z-index	Number, auto
	(Work with absolute/fixed/relative position only)
		div{position:fixed;
		z-index:20;
		background-color:aqua;}

CSS float property:

Note: Absolutely positioned elements ignore the float property.

Method	Values	Example

float	none|left|right	img{float:left;
		width:50px;height:50px;}

Advance CSS

CSS Image:

Method	Values	Example

Rounded Images	border-radius
	border-top-right-radius
	border-top-left-radius
	border-bottom-right-radius
	border-bottom-left-radius
		img{
		border: 1px solid blue;
		border-radius:5px;
		width:200px;
		}

Thumbnail Image	Padding	img{
		border: 1px solid blue;
		border-radius:5px;
		width:200px;
		padding:50px;
		}

Responsive Image	max-width:100%; height:auto;	img {max-width:100%; height:auto; }

Transparent Images	opacity : 0(transparent)/1(solid)	img{
		opacity:0.7;}

CSS Gradient:

Method	Values	Example

Linear	background-image: linear-gradient (direction, color-stop1, color-stop2, ...);
	Default : Top to bottom
		.grad
		{
		background-image:linear-gradient(to right,pink,red);
		}
		background-image:
		linear-gradient(0deg,blue,cyan);

Radial	background-image: radial-gradient(*shape size* at *position, start-color, ..., last-color*); **Default : ellipse**	background-image: radial-gradient(circle, red, yellow, green)
Conic	background-image: conic-gradient([from *angle*] [at *position*,] *color* [*degree*], *color* [*degree*], ...); **Default: 0deg**	conic-gradient(red, yellow, green, blue, black) -------------------------------------- conic-gradient(red 0deg, red 90deg, yellow 90deg, yellow 180deg, green 180deg, green 270deg, blue 270deg);

CSS Transition: (Work on any action like :hover)

Method	Values	Example
transition-property: value1,value2,value3....;	Width, height, color etc…	transition-property: width;
transition-delay: value1,value2,value3....;	In time	transition-delay: 2 s
transition-duration: value1,value2,value3....	In Time (*Compulsory Property to use)	transition-duration: 5s

CSS 2D Transforms Methods

Method	Values	Example

translate()	moves an element from its current position	transform: translate(100px, 75px);
		translateX() & translateY()

rotate()	rotates an element (+ve) clockwise or (-ve) counter-clockwise *used to rotate image also	transform: rotate(45deg)

scale()	increases or decreases the size of an element	transform: scale(2,3);
		transform: scaleX(2);
		transform: scaleY(2);

skew()	skews an element along the X and Y-axis by the given angles	transform: skew(50deg, -15deg); transform: skewX(50deg) transform: skewY(50deg)

CSS Animation:

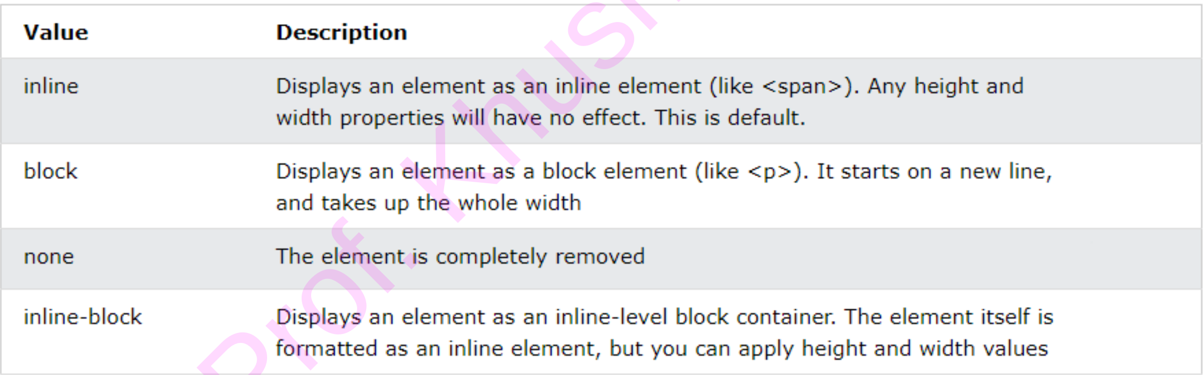
Method	Values	Example

animation-name:	keyframename	animation-name: example;
		animation-delay:2s;
		animation-direction:normal;
		animation-duration: 4s;
		animation-iteration-count: 3

animation-delay	time	
animation-direction	normal|reverse|alternate|alternate-reverse	
animation-duration	time	
animation-iteration-count	Number/infinite	

@keyframes	@keyframes animationname {keyframes-selector {css-styles;}}	@keyframes example {
		from {background-color: lightskyblue;}
		to {background-color: blueviolet;}
		}
		------------------------------- 	
		@keyframes example {
		0% {background-color:lightblue; left:0px; top:0px;}
		50% {background-color:blue; left:200px; top:200px;}
		100% {background-color:lightblue; left:0px; top:0px;} }

CSS Display:



Value	Description
inline	Displays an element as an inline element (like <span>). Any height and width properties will have no effect. This is default.
block	Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width
none	The element is completely removed
inline-block	Displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values

CSS Button:

Properties: background-color, border, color, padding, text-align, display, font-size,

cursor: pointer/auto/zoom-in etc..

Hoverable button – must include→ transition-duration:0.5s;

CSS Tooltip: Properties are: display,visibility, width, background-color, color, text-align, position etc….

Note: Use **visibility:hidden** in main class and **visibility:visible** in hover class

For example

```
.tooltip{ display:inline-block;…}
.text{ visibility:hidden; ………………..}
.tooltip:hover  .text{visibility:visible;……………}
```

CSS Multiple column:

Method	**Values**	**Example**
column-gap	Px	column-gap:20px;
column-rule-style	Solid/dashed/double/dotted	column-rule-style:dashed;
column-rule-color	Color name	column-rule-color:pink;
column-count	number	column-count: 3;
Column-rule-width	px	Column-rule-width:3px

CSS Variable:

Define variable **globally** in side :root{} class
:root{ --b: blue; }
Use of variable inside var() function
P{ color:var(--b) }

All the elements of the document can use the variable.

Define variable **locally** and use it
P{ --b: blue; color:var(--b) }
It has a scope only for the p element.

CSS pagination:

```
.pagination
          {
            display:inline-block;
            }
.pagination a{
          color:black;
          text-decoration:none; /*if not used none then it shows underline.*/
          padding:20px;
          }
a.active {
           background-color: lightgreen;
           color: white;
          }
a:hover:not(.active) {background-color: lightblue;}
```

```
          <div class="pagination">
                    <a href="#">&laquo</a>
                    <a href="#">1</a>
                    <a class="active" href="#">2</a>
                    <a href="#">3</a>
                    <a href="#">&raquo</a>
          </div>
```

Rounded Active & Hoverable Buttons: Add the border-radius property if you want a rounded "active" and "hover" button.

Bordered pagination: Use the border property to add borders to the pagination.

Pagination size: Change the size of the pagination with the font-size property.

Media Queries

Must Include : <meta name="**viewport**" content="width=device-width, initial-scale=1.0/>

@media not|only *mediatype* and *(mediafeature* and|or|not *mediafeature)* {*CSS-Code;*}

not: The not keyword inverts the meaning of an entire media query.

only: The only keyword prevents older browsers that do not support media queries with media features from applying the specified styles. **It has no effect on modern browsers.**

and: The and keyword combines a media feature with a media type or other media features.

They are all optional. However, if you use **not** or **only**, you must also specify a media type.

Examples:

@media only screen and (orientation: landscape/portrait){…………}

@media only screen and (max-width: 600px){…………..}

@media only screen and (min-width: 600px) and (max-width:800px) {………….}

Example:

```
<style>
body{ background-color:lightgreen; }

@media only screen and (orientation:landscape){
body{background-color:pink;}
}
</style>
```