**CSS Important Points**

* Background shorthand property:

**background-color background-image background-repeat background-attachment background-position**

Ex: background: red url(“1.jpg”) no-repeat fixed top left;

* Border shorthand property: if **border-style** is not set other values become ineffective.

1. The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: **thin, medium, or thick:**
2. We can specify different border for each side.

border-top-style:

border-right-style:

border-bottom-style:

border-left-style

**border-width border-style(required) border-color:**

Ex: 2px solid(required) black;

Ex: thin dotted(required)black;

**Rounded Borders:**

Border-radius is used to add rounded borders to the element

Border-radius: 5px;

We can add different radius for all 4 corners of the border.

* Margin property in CSS can be specified for each side of the element.

margin-top:

margin-right:

margin-bottom:

margin-left:

All the margin properties can have the following values:

1. Auto: browser calculates the margin
2. Length specified in px, pt, cm etc.
3. %: with respect to the width of the containing element.
4. Inherit: margin is inherited from the parent element.

**Negative values are allowed.**

**Shorthand margin property:**

Ex: margin: 25px 30px 75px 100px;

**Margin Collapse:** Top and bottom margins sometimes collapse into a single margin, which is greater of the two. Ex: If bottom margin of upper element is 50px and top margin of lower element is 20px. The vertical margin in total will not be equal to 50px+20px=70px, rather it will be equal to 50px, which is greatest of the two.

* Padding is used to create space between the element content and the border.

1. Padding can be specified for each side of the element.

padding-top:

padding-right:

padding-bottom:

padding-left:

1. Padding can have following values:

Length, %, inherit.

**Shorthand padding property:**

Padding: 20px 30px 30px 40px;

**Padding and Element Width:**

Padding adds to the total width of the element. To maintain the total width of the element, we use box-sizing property. This causes the width to remain same, even if the padding is increased, the content area space decreases.

box-sizing: border-box;

* Height and Width property do not include padding, border or margin. It sets the area inside the padding, border or margin of the element.

1. Height and width have the following values:

Auto, length (in px, cm etc.), %, initial, inherit.

1. max-width property is used to set maximum width of an element.

Values can be length in px, cm or % etc. or set to none.

(when the size of window decreases than the max-width, the width reduces according to the screen size. When size of window is greater than the max-width, then width of the element is equal to max-width.)

**If for an element we use both width and max-width property and the width vale is greater than max-width, then max-width is used.**

* **CSS BOX MODEL**

Css box model is a box that wraps around every HTML element. It consists of : margins, borders, padding and actual content.

**When we set the width and height properties of an element with CSS, you just set the width and height of the content area. To calculate the full size of an element, you must also add padding, borders and margins.**

Total width of an element should be calculated like this:

*Total width= width + left padding + right padding + left border + right border + left margin + right margin.*

*Total height= height + top padding + bottom padding + top border + bottom border + top margin + bottom margin.*

* **Css Outline**

An outline drawn outside the borders.

It has following properties:

1. outline-style
2. outline-color: *name, HEX(#ff00ff), RGB rgb (255,0,0), HSL, invert, ensures that outline is visible, regardless of color background.*
3. outline-width: *thin, medium, thick, specific size (in px, cm etc.)*
4. outline-offset:
5. outline

**Outline may overlap other content. The outline is not a part of the element’s dimensions; the element’s total width and height is not affected by the width of the outline.**

1. **Outline-style** can have one of the following values:

Dotted, dashed, solid, double etc.

**If outline-style property is not set other outline properties will not have any effect.**

1. **Outline shorthand**

Outline-width outline-style(required) outline-color

Ex: outline: 4px solid black;

1. **Outline-offset**

This adds space between an outline and the border of an element. The space between an element and its outline is transparent.

Ex: outline-offset: 15px;

* **CSS Text**

1. Color property is used to add color to text, it can be done using color name, rgb, hex, hsl.
2. If you define the color property, you must also define the background-color.
3. Text-align sets horizontal alignment of a text. Text can be right, centered or justified.

When text-align property is set to “justify”, each line is stretched so that every line has equal width and left and right margins are straight.

1. Direction and Unicode-bidi change the direction of an element.
2. Vertical-align aligns element vertically.

**Values for vertical-align:**

Vertical-align: baseline;

vertical-align: text-top;

vertical-align: text-bottom;

vertical-align: sub;

vertical-align: sup;

1. Text-decoration property sets/removes the decorations from the text.

**Text-decoration: none;** is often used to remove underlines from links.

**Text-decoration: overline;** line over text.

**Text-decoration: line-through;** line through the text.

**Text-decoration: underline;** underline the text.

1. Text-transform is used to specify uppercase and lowercase letters in text.

Values for text-transform are:

**Text-transform: uppercase;**

**Text-transform: lowercase;**

**Text-transform: capitalize; capitalizes first letter of every word.**

1. Text-indentproperty is used to specify the indentation of the first line of a text:

**Text-indent: 50px;**

1. Letter-spacing property specifies the space between the characters in a text.

**Letter-spacing: 3px;**

Letter-spacing value can also be negative. **Letter-spacing: -3px;**

1. Line-height specifies the spaces between lines.

**Line-height: 0.8;**

**Line-height: 1.8;**

1. Word-spacing property is used to specify the space between the words in a text.

**Word-spacing: 10px;**

**Word-spacing: -5px;** (Negative values are possible)

1. White-space specifies how white-space inside an element is handled.

**White-space: nowrap;**

1. Text-shadow adds shadow to text.

We specify the **horizontal shadow (2px)** and the **vertical shadow (2px):**

**Text-shadow: 2px 2px;**

**Text-shadow: 2px 2px red;** (this adds color to the shadow)

**Text-shadow: 2px 2px 5px red;** (adds blur effect to the shadow)

* **CSS Fonts**

**Generic Font families:** Serif, Sans-Serif, Monospace, Cursive, Fantasy.

**Font-family** property is used to specify the font of a text.

font-family property should have several font names as fallback, start with font you want and end with generic family.

**Example:** font-family: “Times New Roman”, Times, serif;

**Web Safe Fonts:** are universally installed across all browsers and devices.

**Fallback Fonts:** similar backup fonts, if first doesn’t work, next one, in the end generic font.

**Font-style:** it has three values: **normal, italic and oblique.**

**Font-weight:** specifies the weight of the font: **normal, lighter, bold, 900** (number).

**Font-variant:** property specifies whether a text should be displayed in small-caps font. In small caps all lowercase letters are converted to uppercase letters. However, the converted uppercase appears to be smaller than the original uppercase text.

**Example:** font-variant: normal; font-variant: small-caps;

**Font-size:** sets the size of the text. Font size can be:

Absolute: sets the text to specified size

Relative: sets the size relative to surrounding elements.

**Example:** font-size: 40px;(you can use zoom tool to resize the paragraph)

Font-size: 2.5em;(1em=16px)(able to adjust text size in all browsers)

1em is current font size.

**Responsive Font Size:** vw = viewport width

Viewport is browser window size. 1vw = 1% of viewport width. If viewport is 50 cm width.

**Font shorthand:**

Font-style font-variant font-weight font-size/line-height font-family.

Font size and font family are required. If one of the values are missing default is used.

**Example:** font: italic small-caps bold 20px/5px Georgia, Serif;

* **CSS Links**

Links are styled depending on their states. The states of the links are defined using pseudo classes.

Four link states are:

1. a:link;
2. a:visited;
3. a:hover;
4. a:active;

text-decoration property is used to remove underline from link.

Example: text-decoration: none;

* **CSS List**

List-style-type: circle;

List-style-type: square;

List-style-type: upper-roman;

List-style-type: lower-alpha;

List-style-image: url(“abc.gif”);

List-style-position: inside; (bullet points are inside the list item).

List-style-position: outside;

List style: square inside url(“abc.gif”);

* **CSS Tables**

**Full width table:**

Width: 100%;

**Collapse table borders:**

Border-collapse: collapse;

**Text-align**

By default the content of <th> is center aligned and the content of <td> is left aligned.

**Vertical align**

It sets the vertical alignment, by default the vertical alignment of the content in a table is middle.

**Hover able Table**

Tr:hover{background-color: #ff0000}

**Striped Table**

tr:nth-child(even) {background-color: red;}

**Responsive Table:**

style=”overflow-x:auto;”

* **CSS Display Property**

Default display value is block/inline.

**Block-level elements:**

Block level elements always start with a new line and it take full width available.

**Inline elements:**

Inline element does not start on a new line and only takes width as much necessary.

**To hide:**

display:none; (**the page will be displayed as if the element is not there**)

visibility:hidden; **also hides the element, however the element still takes the same space as before.**

Setting display property only changes how an element is displayed, not what kind of element it is. So, an inline element with display: block is not allowed to have another block element inside it.

* **CSS position property**

1. Static
2. relative
3. fixed:

**The element is removed from the normal document flow, and no space is created for the element in the page layout.**

1. absolute
2. sticky

Then we position elements using top, bottom, left, right properties. These properties won’t work if position property is not set.

**Overlapping Elements:**

Z-index: can have positive and negative values.

**Important: if we don’t specify any position property then the element will be displayed one after other. When we specify position : relative for parent element then that element will be position relative to its normal position by setting top left bottom right properties. For its child element if position: absolute is set, it will be positioned according to the position of the parent whose position is relative. If the parent has no position property set then the child’s position will be with respect to document’s body.**

* **CSS overflow property**

controls what happens to text that too big to fit.

visible - Default. The overflow is not clipped. The content renders outside the element's box

hidden - The overflow is clipped, and the rest of the content will be invisible

scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content

auto - Similar to scroll, but it adds scrollbars only when necessary

overflow-x specifies what to do with the left/right edges of the content.  
overflow-y specifies what to do with the top/bottom edges of the content.

**The overflow property only works for block elements with a specified height**

* **CSS Float and Clear**

The CSS float property specifies how an element should float.

The CSS clear property specifies what elements can float beside the cleared element and on which side.

The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.

the float property can be used to wrap text around images.

left - The element floats to the left of its container

right - The element floats to the right of its container

none - The element does not float (will be displayed just where it occurs in the text). This is default

inherit - The element inherits the float value of its parent

The clear property can have one of the following values:

* none - Allows floating elements on both sides. This is default
* left - No floating elements allowed on the left side
* right- No floating elements allowed on the right side
* both - No floating elements allowed on either the left or the right side
* inherit - The element inherits the clear value of its parent

**When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left.**

* **CSS 2D Transform**

Transform CSS property. With CSS transform property we use 2D transformation methods:

* 1. translate(): translate() moves the element from its current position according to x and y axis parameters.

Example:

div {

transform: translate(50px, 100px);

}

* 1. rotate(): The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.  
     Example:   
     div {  
       transform: rotate(20deg);  
     }