CSS CheatSheet

CODE HELP

In this Cheatsheet, we will cover the basics of CSS properties. We will provide examples to help you understand how these properties work and how to use them in your own web development projects. Whether you are a beginner or an experienced developer, this PDF can serve as a useful reference guide.

CSS

CSS (**Cascading Style Sheets**) is a stylesheet language used for describing the look and formatting of a document written in HTML (Hypertext Markup Language).

CSS is used to define the layout, font, color, and other visual aspects of a webpage, and it helps to separate the content of a webpage from its presentation. CSS allows you to apply styles to multiple HTML elements at once, and it makes it easy to maintain and update the styling of a webpage.

You can use CSS to specify styles for different devices, such as desktop computers, tablets, and mobile phones, which makes it easier to create responsive designs that look good on any device. To use CSS, you can include a stylesheet in your HTML document using a <link> element, or you can use inline styles in your HTML elements using the style attribute.

You can also use CSS to style elements in other documents, such as XML or SVG, and you can use CSS in combination with other technologies, such as JavaScript, to create dynamic and interactive webpages.

FONT PROPERTIES IN CSS

The font has many properties that you can change, such as its face, weight, and style, which allow you to alter the appearance of your text.

- **Font-Family:** Specifies the font family of the text.
- **Font:** A shorthand property for specifying the font-style, font-variant, font-weight, font-size/line-height, and font-family properties all at once.
- **Font-Size:** Specifies the font size of the text.
- **Font-Weight:** Specifies the weight of a font.
- **Font-Style:** Specifies the font style of the text.
- **Font-Variant:** Specifies whether or not the text is displayed in a small-caps font.

```
p {
    font-family: Times, serif, Arial, Helvetica, sans-serif;
    font: 15px Helvetica, sans-serif, Arial;
    font-size: 15px;
    font-weight: bold;
    font-style: italic;
    font-variant: small-caps;
}
```

TEXT PROPERTIES IN CSS

CSS has a lot of properties for formatting text.

- **Text-Align:** It specifies the horizontal alignment of text.
- **Text-Decoration:** It specifies the decoration added to text.
- **Letter-Spacing:** It increases or decreases the space between characters in a text.
- **Text-Transform:** It controls the capitalization of text.
- **Word-Spacing:** It increases or decreases the space between words in a text.
- **Text-Indent:** It specifies the indentation of the first line in a text-block.
- **Line-Height:** It sets the line height.
- **Text-Shadow:** It adds shadow to the text.

```
p {
    text-align: center;
    text-decoration: underline;
    letter-spacing: 5px;
    text-transform: uppercase;
    word-spacing: 8px;
    text-indent: 40px;
    line-height: 40%;
    text-shadow: 4px 4px #ff0000;
}
```

BACKGROUND PROPERTIES IN CSS

The background properties allow you to customize the color, image, position, size, and other aspects of the document's background. As the name implies, these properties affect the background of the document.

- **Background-Image:** It specifies one or more background images for an element.
- **Background-Size:** It specifies the size of the background images.
- **Background-Position:** It specifies the position of a background image.
- **Background-Repeat:** It sets, how a background image will be repeated.
- **Background-Color:** It specifies the background color of an element.
- **Background-Attachment:** It sets whether a background image scrolls with the rest of the page, or is fixed.
- **Background-Origin:** It specifies the origin position of a background image.
- **Background:** A shorthand property for all the background-* properties.

```
body {
      background-image:  url('codeHelp.png');
      background-size:  auto;
      background -position:  center;
      background -repeat:  no-repeat;
      background -color:  #ffffff;
      background -attachment:  fixed;
      background -origin:  content-box;
      background :  url('codeHelp.png');
}
```

BORDERS PROPERTIES IN CSS

The border properties enable you to alter the style, radius, color, and other characteristics of the buttons or other elements within the document.

- **Border-Width:** It sets the width of the four borders.
- **Border-Style:** It sets the style of the four borders.
- **Border-Radius:** A shorthand property for the four border-*-radius properties.
- **Border-Color:** It sets the color of the four borders.

- **Border:** A shorthand property for border-width, border-style and border-color.

- **Border-Spacing:** It sets the distance between the borders of adjacent cells.

```
div {
    border-width: 4px;
    border-style: solid;
    border-radius: 4px;
    border-color: #000000;
    border: 20px dotted coral;
    border-spacing: 20px;
}
```

BOX MODEL PROPERTIES IN CSS

The CSS box model is a structure that encloses every HTML element, and is composed of margins, borders, padding, and the element's content. This model is utilized to design and arrange the layout of web pages.

- **Padding:** A shorthand property for all the padding-* properties.

- **Visibility:** It specifies whether or not an element is visible.

- **Display:** It specifies how a certain HTML element should be displayed.

- **Height:** It sets the height of an element.
- **Width:** It sets the width of an element.
- **Float:** It specifies whether an element should float to the left, right, or not at all.
- **Clear:** It specifies what should happen with the element that is next to a floating element.
- **Margin:** It sets all the margin properties in one declaration.
- **Overflow:** It specifies what happens if content overflows an element's box.

```
p {
    padding: 10px 20px 10px 20px;
    visibility: hidden;
    display: inline-block;
    height: auto;
    width: 100px;
    float: right;
    clear: left;
    margin: 20px 10px 20px 10px;
    overflow: scroll;
}
```

COLORS PROPERTIES IN CSS

The color property can be used to add color to various objects.

- **Color:** It sets the color of text.
- **Outline-Color:** It sets the color of an outline.
- **Caret-Color:** It specifies the color of the cursor (caret) in inputs, textareas, or any element that is editable.
- **Opacity:** It sets the opacity level for an element.

```
{
    color: rgb(0, 0, 0);
    outline-color: #000000;
    caret-color: coral;
    opacity: 0.8;
}
```

LAYOUT PROPERTIES IN CSS

It defines the appearance of the content within a template.

- **Box-Align:** It specifies how an element aligns its contents across its layout in a perpendicular direction.
- **Box-Direction:** It specifies whether a box lays out its contents normally (from the top or left edge), or in reverse (from the bottom or right edge).
- **Box-Flex:** This property specifies how a box grows to fill the box that contains it, in the direction of the containing box's layout.
- **Box-Orient:** It sets whether an element lays out its contents horizontally or vertically.
- **Box-Sizing:** It allows us to include the padding and border in an element's total width and height.
- **Box-Pack:** This property specifies how a box packs its contents in the direction of its layout.
- **Min-Width:** It sets the minimum width of an element.
- **Max-Width:** It sets the maximum width of an element.
- **Min-Height:** It sets the minimum height of an element.
- **Max-Height:** It sets the maximum height of an element.

```
{
    box-align:  start;
    box-direction:  normal;
    box-flex:  normal;
    box-orient:  inline;
    box-sizing:  margin-box;
    box-pack:  justify;
    min-width:  200px;
    max-width:  400px;
    min-height: 100px;
    max-height:  1000px;
}
```

TABLE PROPERTIES IN CSS

Table properties allow you to customize the appearance of tables in a document, including options like border spacing, table layout, and captions.

- **Border-Spacing:** It sets the distance between the borders of adjacent cells.
- **Border-Collapse:** It sets whether table borders should collapse into a single border or be separated.
- **Empty-Cells:** It specifies whether or not to display borders and background on empty cells in a table.
- **Caption-Side:** It specifies the placement of a table caption.

- **Table-Layout:** It defines the algorithm used to layout table cells, rows, and columns.

```
{
    border-spacing:  4px;
    border-collapse:  separate;
    empty-cells:  show;
    caption-side:  bottom;
    table-layout:  auto;
}
```

COLUMNS PROPERTIES IN CSS

These properties are specifically applied to the columns of a table and are used to enhance its appearance.

- **Column-Gap:** It specifies the gap between the columns.
- **Column-Rule-Width:** It specifies the width of the rule between columns.
- **Column-Rule-Color:** It specifies the color of the rule between columns.
- **Column-Rule-Style:** It Specifies the style of the rule between columns.
- **Column-Count:** It specifies the number of columns an element should be divided into.
- **Column-Span:** It specifies how many columns an element should span across.

- **Column-Width:** It specifies the column width.

```
{
    column-gap:  4px;
    column-rule-width:  medium;
    column-rule-color:  #000000;
    column-rule-style:  dashed;
    column-count:  20;
    column-span:  all;
    column-width:  4px;
}
```

LIST & MARKER PROPERTIES IN CSS

The list and marker properties can be used to customize the appearance of lists in a document.

- **List-Style-Image:** It specifies an image as the list-item marker.
- **List-Style-Position:** It specifies the position of the list-item markers (bullet points).
- **List-Style-Type:** It specifies the type of list-item marker.
- **Marker-Offset:** It allows you to specify the distance between the marker and the text relating to that marker.

```
{
    list-style-image:  url('codeHelp.png');
    list-style-position:  10px;
    list-style-type:  square;
    marker-offset:  auto;
}
```

ANIMATION PROPERTIES IN CSS

CSS animations enable the creation of animated transitions or the animation of other media elements on a web page.

- **Animation-Name:** It specifies a name for the @keyframes animation.
- **Animation-Delay:** It specifies a delay for the start of an animation.
- **Animation-Duration:** It specifies a delay for the start of an animation.
- **Animation-Timing-Function:** It specifies the speed curve of an animation.
- **Animation-Iteration-Count:** It specifies the number of times an animation should be played.
- **Animation-Fill-Mode:** It specifies a style for the element when the animation is not playing (before it starts, after it ends, or both).

- **Animation-Play-State:** It specifies whether the animation is running or paused.

- **Animation-Direction:** It specifies whether the animation should play forward, backward, or alternate back and forth between playing the sequence forward and backward.

```
{
    animation-name: anime;
    animation-delay: 4ms;
    animation-duration: 10s;
    animation-timing-function: ease;
    animation-iteration-count: 5;
    animation-fill-mode: both;
    animation-play-state: running;
    animation-direction: normal;
}
```

TRANSITION PROPERTIES IN CSS

Transitions allow you to specify how an element will change from one state to another.

- **Transition-Property:** It specifies the name of the CSS property the transition effect is for.

- **Transition-Delay:** It specifies when the transition effect will start.

- **Transition-Duration:** It specifies how many seconds or milliseconds a transition effect takes to complete.

- **Transition-Timing-Function:** It specifies the speed curve of the transition effect.

```
{
    transition-property:  none;
    transition-delay:   4ms;
    transition-duration:   10s;
    transition-timing-function:   ease-in-out;
}
```

CSS FLEXBOX

Flexbox is a CSS layout system that makes it easy to align and distribute items within a container using rows and columns. It allows items to "flex" and adjust their size to fit the available space, making responsive design simpler to implement. Flexbox makes formatting HTML elements more straightforward and efficient.

PARENT PROPERTIES:

- **Display:** It specifies how a certain HTML element should be displayed.
- **Flex-Direction:** It specifies the direction of the flexible items.
- **Flex-Wrap:** It specifies whether the flexible items should wrap or not.
- **Flex-Flow:** It is a shorthand property for the flex-direction and the flex-wrap properties.

- **Justify-Content:** It specifies the alignment between the items inside a flexible container when the items do not use all available space.

- **Align-Items:** It specifies the alignment for items inside a flexible container.

- **Align-Content:** It specifies the alignment between the lines inside a flexible container when the items do not use all available space.

```
{
    display: flex;
    flex-direction: row | row-reverse | column | column-reverse;
    flex-wrap: nowrap | wrap | wrap-reverse;
    flex-flow: column wrap;
    justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly | start | end | left | right ... + safe | unsafe;
    align-items: stretch | flex-start | flex-end | center | baseline | first baseline | last baseline | start | end | self-start | self-end + ... safe | unsafe;
    align-content: flex-start | flex-end | center | space-between | space-around | space-evenly | stretch | start | end | baseline | first baseline | last baseline + ... safe | unsafe;
}
```

CHILD PROPERTIES:

- **Order:** It sets the order of the flexible item, relative to the rest.
- **Flex-Grow:** It specifies how much the item will grow relative to the rest.
- **Flex-Shrink:** It specifies how the item will shrink relative to the rest.
- **Flex-Basis:** It Specifies the initial length of a flexible item.
- **Align-Self:** It specifies the initial length of a flexible item.

```
{
    order:  2;                          /* By default it is 0 */
    flex-grow:  5;                      /* By default it is 0 */
    flex-shrink:  4;                    /* By default it is 1 */
    flex-basis: | auto;                 /* By default it is auto */
    align-self:  auto | flex-start | flex-end | center | baseline | stretch;
    flex:  none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ];
}
```

CSS GRID

The Grid layout is a 2-dimensional system in CSS that allows for the creation of complex and responsive web design layouts with consistent results across different browsers. It makes it easier to build these types of layouts.

PARENT PROPERTIES:

- **Display:** It specifies how a certain HTML element should be displayed.
- **Grid-Template-Columns:** It specifies the size of the columns, and how many columns in a grid layout.
- **Grid-Template-Rows:** It specifies the size of the columns, and how many columns in a grid layout.
- **Grid-Template:** It is a shorthand property for the grid-template-rows, grid-template-columns and grid-areas properties.
- **Column-Gap:** It specifies the gap between the columns.
- **Row-Gap:** It specifies the gap between the grid rows.
- **Grid-Column-Gap:** It specifies the size of the gap between columns.
- **Grid-Row-Gap:** It specifies the size of the gap between rows.
- **Gap:** It is a shorthand property for the row-gap and the column-gap properties.

- **Grid-Gap:** It is a shorthand property for the grid-row-gap and grid-column-gap properties.

- **Align-Items:** It specifies the alignment for items inside a flexible container.

- **Justify-Content:** It specifies the alignment between the items inside a flexible container when the items do not use all available space.

- **Align-Content:** It specifies the alignment between the lines inside a flexible container when the items do not use all available space.

- **Grid-Auto-Columns:** It specifies a default column size.

- **Grid-Auto-Rows:** It specifies a default row size.

- **Grid-Auto-Flow:** It specifies how auto-placed items are inserted in the grid.

```
{
    display:  grid | inline-grid;
    grid-template-columns:  10px 10px 10px;
    grid-template-rows:  5px auto 10px;
    grid-template:  none | <grid-template-rows> / <grid-template-
    columns>;
    column-gap:  <line-size>;
    row-gap:  <line-size>;
    grid-column-gap:  <line-size>;
    grid-row-gap:  <line-size>;
    gap:  <grid-row-gap> <grid-column-gap>;
    grid-gap:  <grid-row-gap> <grid-column-gap>;
    align-items:  start | end | center | stretch;
    justify-content:  start | end | center | stretch | space-around | space-
    between | space-evenly;
    align-content:  start | end | center | stretch | space-around | space-
    between | space-evenly;
    grid-auto-columns:  <track-size>;
    grid-auto-rows:  <track-size>;
    grid-auto-flow:  row | column | row dense | column dense;
}
```

CHILD PROPERTIES:

- **Grid-Column-Start:** It specifies where to start the grid item.
- **Grid-Column-End:** It specifies where to end the grid item.
- **Grid-Row-Start:** It specifies where to start the grid item.
- **Grid-Row-End:** It specifies where to end the grid item.
- **Grid-Column:** It is a shorthand property for the grid-column-start and the grid-column-end properties.
- **Grid-Row:** It is a shorthand property for the grid-row-start and the grid-row-end properties.
- **Grid-Area:** It either specifies a name for the grid item, or this property is a shorthand property for the grid-row-start, grid-column-start, grid-row-end, and grid-column-end properties.
- **Align-Self:** It specifies the alignment for selected items inside a flexible container.

```
{
    grid-column-start:  <number> | <name> | span <number> | span <name>
    | auto;
    grid-column-end:  <number> | <name> | span <number> | span <name> |
    auto;
    grid-row-start:  <number> | <name> | span <number> | span <name> |
    auto;
    grid-row-end:  <number> | <name> | span <number> | span <name> |
    auto;
    grid-column:  <start-line> / <end-line> | <start-line> / span <value>;
    grid-row:  <start-line> / <end-line> | <start-line> / span <value>;
    grid-area:  <name> | <row-start> / <column-start> / <row-end> /
    <column-end>;
    align-self:  start | end | center | stretch;
}
```