GLS University

Faculty of Computer Applications & Information Technology

0301103 Introduction to HTML5, CSS and Javascript

UNIT – 3

BCA

CSS Introduction

- Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML.
- CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.
- It sets the background color, font-size, font-family, color, … etc property of elements in a web pages.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.
- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

Advantages of CSS

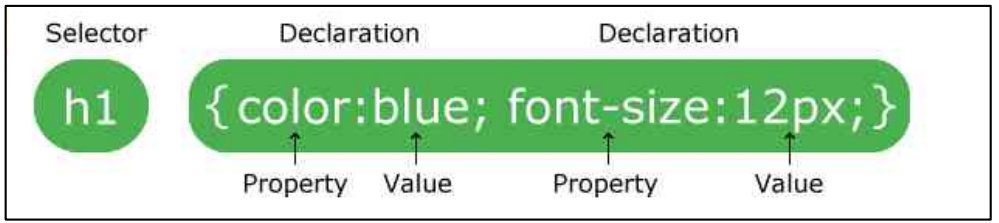
- Saves Time
- Help to Make Spontaneous and Consistent Changes
- Improves Page Loading Speed
- Better Device Compatibility
- Ability to Re-Position
- Makes the Search Engine Better Crawl Your Web Pages

Disadvantages of CSS

- Cross-Browser Issues – This is simply due to the fact that CSS works differently on different browsers.
- Confusion Due to Its Many Levels
- Vulnerable - It would only require a read/write access to the intended website to override the changes.

CSS Syntax

- A CSS rule-set consists of a selector and a declaration block



- **Selector -** points to the HTML element you want to style.
- **Declaration –** block contains one or more declarations separated by semicolons.
- **Declaration** includes a CSS property name and a value, separated by a colon.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

CSS Class

- The HTML class attribute is used to define equal styles for elements with the same class name.
- So, all HTML elements with the same class attribute will get the same style.
- The class attribute can be used on any HTML element.
- The class name is **case sensitive!**
- HTML elements can have more than one class name, each class name must be separated by a space.
- Different tags can have the same class name and thereby share the same style.
- **Class selector is a name preceded by a full stop (“.”)**
- **Eg: .title_style1**

- Two ways by which we can write css
- 1 by class
- 2 By id.

Class and ID Selector

- Each element can have only one ID
- Each page can have only one element with that ID
- **ID selector is a name preceded by a hash character (“#”).**

Difference: class vs id selector

- Each element can have more than one Class
- Each page can have multiple elements with same class
- **class selector is a name preceded by a dot character (“.”).**

- Each element can have only one ID
- Each page can have only one element with that ID
- **ID selector is a name preceded by a hash character (“#”).**

CSS Comments

- */* This is commented text */*
- This allows you to enter notes into CSS that will not be interpreted.

Types of CSS – Inline CSS

- It contains the CSS property in the body section attached with element is known as inline CSS.
- This kind of style is specified within an HTML tag using style attribute.

Eg:

```
<p style = "color:#009900;
            font-size:50px;
            font-style:italic;
            text-align:center;">
    GLS University </p>
```

Types of CSS - Internal or Embedded CSS

- This can be used when a single HTML document must be styled uniquely.
- The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file.
- It is specified within <style> tag.
- Eg:

```
<style>
        .main
        {
            text-align:center;
        }
</style>
```

Types of CSS - External CSS

- External CSS contains separate CSS file which contains only style property with the help of tag attributes.
- CSS property written in a separate file with .css extension and should be linked to the HTML document using link tag.
- This means that for each element, style can be set only once and that will be applied across web pages.

CSS Properties and Attributes

all: initial|inherit|unset;

- **Initial - Changes all the properties applied to the element or the element's parent to their initial value**
- **Inherit** - Changes all the properties applied to the element or the element's parent to their parent value
- **Unset** - Changes all the properties applied to the element or the element's parent to their parent value if they are inheritable or to their initial value if not

CSS Properties and Attributes - Background

Used to set background styles for HTML elements.

background	Shorthand for setting all the properties
background-color	Sets back color for element. (colorname/ color value/ transperent)
background-attachment	Specifies if background image is fixed or scroll. (scroll/fixed)
background-position	Specifies initial position of background image. (left/right/center/top/bottom/ x% y%, x-position, y-position)
background-image	Specifies backgound image. (web address/url)
background-repeat	Specifies whether backgound image is reapeated or not. (repeat/repeat-x/repeat-y/no-repeat)

CSS Properties and Attributes - TEXT

color	Specifies text(foreground) color for text (text color/ hex value)
letter-spacing	Spacing between characters. (normal/ px/in/pt/cm)
line-height	Spacing between two lines. (length/ number/%)
text-align	Alignment of text (left/ right/ center/ justified)
text-decoration	Decorative effects (none/ underline/ overline/ line-through/ blink)
text-indent	Indentation of the first line of the text in a block.
text-shadow	Comma seperated list of shadow effects. (none/ colorname/ colorvalue/ length value)
vertical-align	Vertical positioning of the text. (baseline/ sub/ super/ top/ next-top/ middle/ bottom/ text-bottom/ length/ %)
white-space	Normal/ pre/ nowrap
word-spacing	Space between two consecutive words in a line. (normal/ value)

CSS Properties and Attributes - FONT

font	Defines shorthand for all properties
font-family	Specifies list of font family names
font-size	Size of fonts (xx-small / x-small / small / medium / large / x-large / xx-large / smaller / larger / % val) **If you do not specify a font size, the default size for normal text, like paragraphs, is 16px**
font-stretch	Values can be (normal / wider / narrower / ultra-condensed / extra-condensed / condensed / semi-condensed / semi-expanded / expanded / extra-expanded / ultra-expanded
font-style	Values can be (normal / italic / oblique)
font-variant	Values can be (normal / small-caps)
font-weight	Values can be (normal / bold / bolder / lighter / 100 to 900)

CSS Properties and Attributes-BORDER [TABLE]

border	Shorthand for setting all the properties
border-collapse	sets whether the table borders should be collapsed into a single border
width-height	Width and height of a table are defined by the width and height properties.
text-align	The text-align property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>. By default, the content of <th> elements are center-aligned and the content of <td> elements are left-aligned.

CSS Properties and Attributes - BORDER

vertical-alignment	sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>. By default, the vertical alignment of the content in a table is middle (for both <th> and <td> elements).
padding	To control the space between the border and the content in a table, use the padding property on <td> and <th> elements
border-bottom	Add the border-bottom property to <th> and <td> for horizontal dividers
:hover	Perform some action on mouse over

CSS Properties and Attributes - PADDING

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
- There are properties for setting the padding for each side of an element (top, right, bottom, and left).

CSS Properties and Attributes - PADDING

- CSS has properties for specifying the padding for each side of an element as:

  padding-top / padding-right / padding-bottom / padding-left

All the padding properties can have the following values:

- **length** - specifies a padding in px, pt, cm, etc.
- **%** - specifies a padding in % of the width of the containing element
- **inherit** - specifies that the padding should be inherited from the parent element

- **Negative values are not allowed.**

Shorthand Padding Properties

- To shorten the code, it is possible to specify all the padding properties in one property.
- If the padding property has **4** values, it specifies individual padding values for top, bottom, left and right respectively
- If the padding property has **3** values, it specifies individual padding values for top and bottom respectively and common values for left/right padding
- If the padding property has **2** values, it specifies common values for top/bottom padding and common values for left/right padding
- If the padding property h/as **1** value, it specifies common values for top/bottom/left/right padding

List Properties and Attributes

- In HTML, there are two main types of lists:
- Unordered lists (<ul>) - the list items are marked with bullets
- Ordered lists (<ol>) - the list items are marked with numbers or letters
- The CSS list properties allow you to:
- Set different list item markers for ordered lists
- Set different list item markers for unordered lists
- Set an image as the list item marker
- Add background colors to lists and list items

List Properties and Attributes

Properties	Description
list-style	Sets all the properties for a list in one declaration
list-style-image	Specifies an image as the list-item marker
list-style-position	Specifies the position of the list-item markers (bullet points)
list-style-type	Specifies the type of list-item marker

CSS Positioning Properties

- The position property specifies the type of positioning method used for an element.
- There are five different position values:
  - static
  - relative
  - fixed
  - absolute
  - sticky

position: static

- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page.
- **Eg: div.static**

```
{
    position: static;
    border: 3px solid #73AD21;
}
```

position: relative

- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.
- Other content will not be adjusted to fit into any gap left by the element
- **div.relative**

```
{
    position: relative;
    left:30px;
    border: 3px solid #73AD21;
}
```

position: fixed

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.
- The top, right, bottom, and left properties are used to position the element.
- A fixed element does not leave a gap in the page where it would normally have been located.
- **Eg: div.fixed**

```
{
    position: fixed;
    top: 0; right: 0; width: 300px;
    border: 3px solid #0000ff;
}
```

position: absolute

- An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
- However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.
- **Note: A "positioned" element is one whose position is anything except static.**
- **Eg: div.absolute**

```
{
    position: absolute; top: 80px; right: 0;
    width: 200px; height: 100px;
}
```

position: sticky

- An element with position: sticky; is positioned based on the user's scroll position.
- A sticky element toggles between relative and fixed, depending on the scroll position.
- It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).
- **Eg: div.sticky**

```
{
        position: sticky;
        top: 0;
        padding: 5px;
        background-color: #cae8ca;
        border: 2px solid #4CAF50;
}
```

Overlapping Elements

- When elements are positioned, they can overlap other elements.
- The z-index property specifies the stack order of an element.
- Stack order decides which element should be placed in front of, or behind, the others.
- An element can have a positive or negative stack order.
- An element with greater stack order is always in front of an element with a lower stack order.
- If two positioned elements overlap without a z-index specified, the element positioned last in the HTML code will be shown on top.

Table Properties and Attributes

Properties	Description
border	Sets all the border properties in one declaration
border-collapse	Specifies whether or not table borders should be collapsed
border-spacing	Specifies the distance between the borders of adjacent cells
caption-side	Specifies the placement of a table caption
empty-cells	Specifies whether or not to display borders and background on empty cells in a table
table-layout	Sets the layout algorithm to be used for a table