Ritesh Jha | 16010423076 | SY-IT(B3)
WPL Tutorial - 4 | 05-02-25

Aim: To Learn about Extensible Markup Language (XML) and Extensible Stylesheet Language (XSL)

Theory :

XML (Extensible Markup Language) is a markup language used to store and transport data in a structured way. It is human-readable and machine-readable.

- XML uses tags (like HTML) but is not used for designing web pages.
- XML is self-descriptive (you define your own tags).
- XML is case-sensitive (`<book>` is different from `<Book>`).
- It must have a root element (here, `<book>`).

XML only stores data, but if you want to display XML data nicely, you use XSL (XSLT).

- Transforming XML data into HTML or other formats.
- Styling XML elements like CSS.

Scripts :

1) XML + XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="computer_architecture.xsl"?>
<computer>
  <component>
    <name>CPU</name>
    <function>Processes instructions</function>
  </component>
  <component>
    <name>RAM</name>
    <function>Stores temporary data</function>
```

```
    </component>
    <component>
        <name>Hard Drive</name>
        <function>Stores permanent data</function>
    </component>
</computer>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <html>
            <head>
                <title>Computer Internal Architecture</title>
                <style>
                    body { font-family: Arial, sans-serif; }
                    h2 { color: darkblue; }
                    table { width: 50%; border-collapse: collapse; }
                    th, td { border: 1px solid black; padding: 8px; }
                </style>
            </head>
            <body>
                <h2>Computer Internal Architecture</h2>
                <table>
                    <tr>
                        <th>Component</th>
                        <th>Function</th>
                    </tr>
                    <xsl:for-each select="computer/component">
                        <tr>
                            <td><xsl:value-of select="name"/></td>
                            <td><xsl:value-of select="function"/></td>
                        </tr>
                    </xsl:for-each>
                </table>
            </body>
        </html>
    </xsl:template>
</xsl:stylesheet>
```

2) XML + CSS

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="computer_architecture.css"?>
<computer>
   <component>
      <name>CPU</name>
      <function>Processes instructions</function>
   </component>
   <component>
      <name>RAM</name>
      <function>Stores temporary data</function>
   </component>
   <component>
      <name>Hard Drive</name>
      <function>Stores permanent data</function>
   </component>
</computer>
```

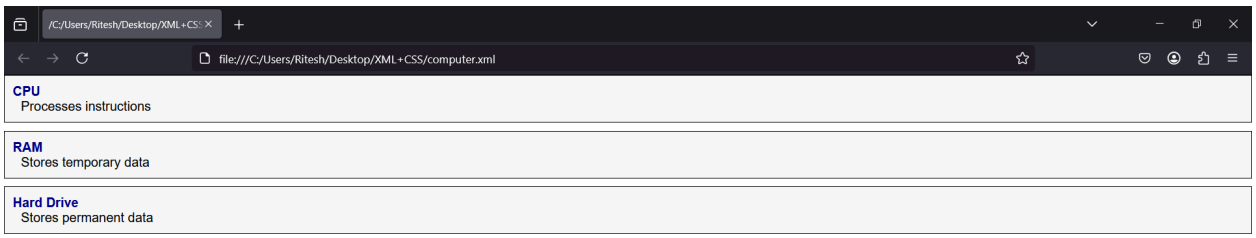
```
computer {
   display: block;
   font-family: Arial, sans-serif;
}

component {
   display: block;
   margin-bottom: 10px;
   padding: 10px;
   border: 1px solid black;
   background-color: #f9f9f9;
}

name {
   display: block;
   font-weight: bold;
   color: darkblue;
}

function {
   display: block;
   margin-left: 10px;
}
```

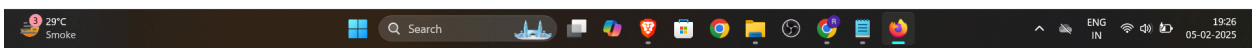
Output :

1) XML + XSL

CPU Processes instructions RAM Stores temporary data Hard Drive Stores permanent data

2) XML + CSS





Outcome :

CO2: Create Web pages using HTML 5 and CSS

Conclusion :

From this tutorial, I learned how XML is used to store and transport data in a structured way. I understood that XML itself does not style or display data, but with XSL, we can transform it into a well-formatted HTML table. I also learned that CSS can be applied to XML to improve its visual presentation. By working with both XSL and CSS, I saw how XML data can be displayed in different ways, making it more readable and user-friendly.