Q1. Explain the structure of an HTML table and the purpose of each of the following elements: <table>, <tr>, <th>, <td>, and <thead>.

Ans: An HTML table is used to display data in a structured format using rows and columns. Tables are created with the <table> element and consist of table rows (<tr>), table headers (<th>), and table data cells (<td>). Additional elements like <thead>, <tbody>, and <tfoot> help organize the table content more clearly.

Purpose of Table Elements

1.   <table> (Table Element):
     The <table> tag defines the start and end of the table. All other table elements like rows and cells are written inside this tag. *Example:*

2.   <table> ... </table>

3.   <tr> (Table Row):
     The <tr> tag is used to define a single row in the table. It contains one or more <th> or <td> elements. *Example:*

4.   <tr> ... </tr>

5.   <th> (Table Header Cell):
     The <th> tag defines a header cell in the table. Text inside <th> is usually bold and centered by default. It represents column headings that describe the type of data in each column. *Example:*

6.   <th>Name</th>

7.   <td> (Table Data)
     The <td> tag defines a data cell — the actual content within the table. Each <td> appears inside a <tr> (row). *Example:*

8.   <td>Nagesh</td>

9.   <thead> (Table Head)
     The <thead> tag groups all the header rows of a table. It helps separate the header section from the body for better readability and styling. *Example:*

10. <thead>

11. <tr>

12. <th>Name</th>

13. <th>Age</th>

14. </tr>

15.   </thead>

Example of a Complete HTML Table

<table border="1">

 <thead>

   <tr>

    <th>Name</th>

    <th>Age</th>

    <th>City</th>

   </tr>

 </thead>

 <tr>

   <td>Nagesh</td>

   <td>23</td>

   <td>Ahmedabad</td>

 </tr>

 <tr>

   <td>Gaurav</td>

<td>17</td>

   <td>Vansda</td>

 </tr>

</table>

Q2. What is the difference between colspan and rowspan in tables? Provide examples.

Ans: In HTML tables, the attributes colspan and rowspan are used to merge cells — either across columns or rows. They help in organizing table data neatly and reducing redundancy.

Attribute	Purpose	Description	Example
colspan	Column Span	Merges two or more columns into a single cell.	<td colspan="2">Merged Columns</td>
rowspan	Row Span	Merges two or more rows into a single cell.	<td rowspan="2">Merged Rows</td>

Example of colspan:

```
<table border="1">
 <tr>
  <th colspan="2">Student Details</th>
 </tr>
 <tr>
  <td>Name</td>
  <td>Age</td>
 </tr>
</table>
```

Example of rowspan:

```
<table border="1">
 <tr>
  <th rowspan="2">Name</th>
  <td>Nagesh</td>
 </tr>
 <tr>
  <td>Gaurav</td>
 </tr>
</table>
```

Q3. Why should tables be used sparingly for layout purposes? What is a better alternative?

Ans: In the early days of web design, HTML tables were often used to create webpage layouts (such as dividing a page into columns and sections). However, this approach is not recommended today because it causes several problems:

1. Poor Accessibility:
   Screen readers and assistive technologies may misinterpret table-based layouts as data tables, making it confusing for visually impaired users.

2. Difficult to Maintain:
   Table-based layouts are complex and hard to modify or redesign, as even small changes require editing multiple rows and columns.
3. Slower Page Loading:
   Tables add extra HTML code, which can increase page size and slow down loading times.
4. Not Responsive:
   Tables do not adapt well to different screen sizes (like mobile devices), making the layout less user-friendly.
5. Mixes Structure and Presentation:
   Tables are meant for displaying tabular data, not for controlling layout. Using them for design breaks the separation between content and styling.

Better Alternative: CSS (Cascading Style Sheets)
The modern and efficient alternative to using tables for layout is CSS.
CSS allows developers to design flexible, responsive, and visually appealing page layouts without affecting the HTML structure.
- CSS Grid and Flexbox are the best layout tools in modern web design.
- They make it easy to create complex layouts that automatically adjust to different screen sizes.