

## Understand table structure?

1. **Browser Support:** Different web browsers may have varying levels of support for CSS features and properties. Major browsers like Chrome, Firefox, Safari, Edge, and Opera generally have good support for modern CSS features, but older versions of browsers may have limited support for newer CSS properties.
2. **CSS Vendor Prefixes:** Some CSS properties require vendor prefixes to work correctly across different browsers. Vendor prefixes are prefixes added to CSS properties to indicate that they are experimental or non-standard features
3. **CSS Resets and Normalization:** CSS resets and normalization techniques are used to ensure consistent rendering of HTML elements across different browsers by removing default browser styles and standardizing CSS properties. Popular CSS reset libraries include Normalize.css and Reset CSS.

**4.Feature Detection:** Feature detection techniques, such as Modernizr.js, can be used to detect the availability of CSS features and provide fallbacks or alternative styles for browsers that do

## 2. Explain cell padding and cell spacing.?

### 1. cellpadding:

- cellpadding is used to specify the amount of space between the content of a table cell and its border.
- This attribute adds padding inside each cell, effectively increasing the distance between the content of the cell and the cell's border.
- The value of cellpadding is specified in pixels (px) or as a percentage (%) of the cell's width. For example, cellpadding="5" or cellpadding="10%".

### cellspacing:

- cellspacing is used to specify the amount of space between adjacent cells in a table.
- The value of cellspacing is specified in pixels (px) or as a percentage (%) of the table's width. For example, cellspacing="5" or cellspacing="10%".

## 3.What are form attributes?

### 1. action:

- Specifies the URL to which the form data will be submitted when the form is submitted.
- Example: <form action="/submit.php">

### 2. method:

- Specifies the HTTP method used when submitting the form data.

- Common values are "get" and "post".
- Example: `<form method="post">`

### 3. target:

- Specifies where to display the response received after submitting the form.
- Common values are "\_blank" (opens in a new window or tab) and "\_self" (opens in the same frame or window).
- Example: `<form target="_blank">`

### 4. name:

- Specifies a name for the form.
- Used to identify the form when submitting multiple forms on a page or **when accessing form data via JavaScript**.
- **Example:** `<form name="contactForm">`

### 5. autocomplete:

- Specifies whether the browser should automatically complete form fields based on the user's input history.
- Common values are "on" (default) and "off".
- Example: `<form autocomplete="off">`

### 6. enctype:

- Specifies the MIME type used to encode form data before submitting it to the server.
- Common values are "application/x-www-form-urlencoded" (default, for simple text data) and "multipart/form-data" (for file uploads).
- Example: `<form enctype="multipart/form-data">`

### 6.target:

- Specifies where to display the response received after submitting the form.
- Common values are "\_blank" (opens in a new window or tab) and "\_self" (opens in the same frame or window).
- Example: `<form target="_blank">`