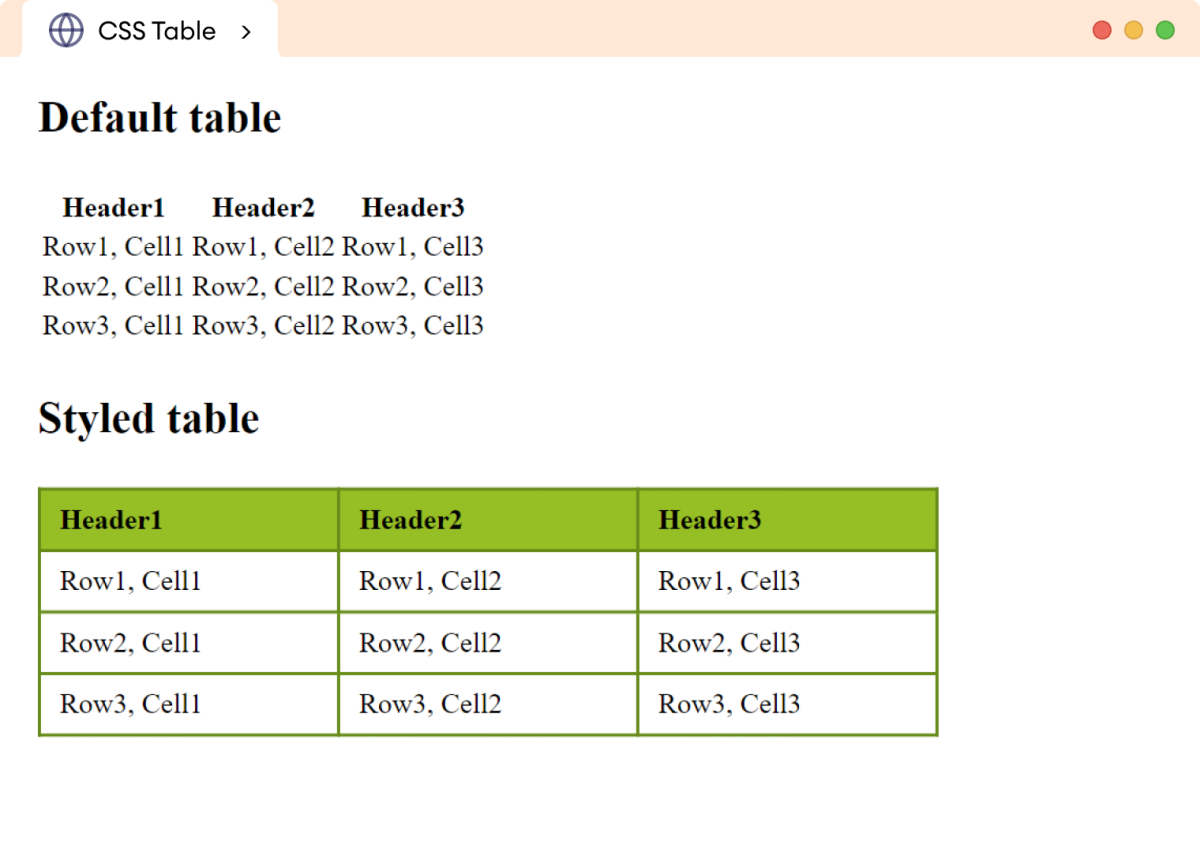
**CSS Table Styling**

**Create a Table**



A table is created with the help of table, tr, th, and td tags. For example,

<table>

<tr>

<th>Name</th>

<th>Location</th>

</tr>

<tr>

<td>James</td>

<td>Chicago</td>

</tr>

<tr>

<td>Robert</td>

<td>New York</td>

</tr>

</table>

**Output**

****

**Style Your Table**

We can add the following styles for the table,

* Table borders
* Collapse table borders
* Border spacing
* Table size
* Table layout
* Horizontal alignment
* Vertical alignment
* Background color
* Table padding
* Hover effect

1.Table borders

<table>

<tr>

<th>Name</th>

<th>Location</th>

</tr>

<tr>

<td>James</td>

<td>Chicago</td>

</tr>

<tr>

<td>Robert</td>

<td>New York</td>

</tr>

<tr>

<td>Charlie</td>

<td>Boston</td>

</tr>

</table>

Css:

/\* applies border to the table \*/

table,th,td {

border: 1px solid black;

}



**Collapse Table Border**

The border-collapse property merges the border between the table cells into a single border. For example,

table {

border-collapse: collapse;

}

**Browser Output**



**Border Spacing**

The border-spacing property specifies the gap between the adjacent cell borders. For example,

table {

border-spacing: 20px;

}

**Browser Output**



**Table Size**

The size of the table can be changed by using the width and height properties. For example,

table {

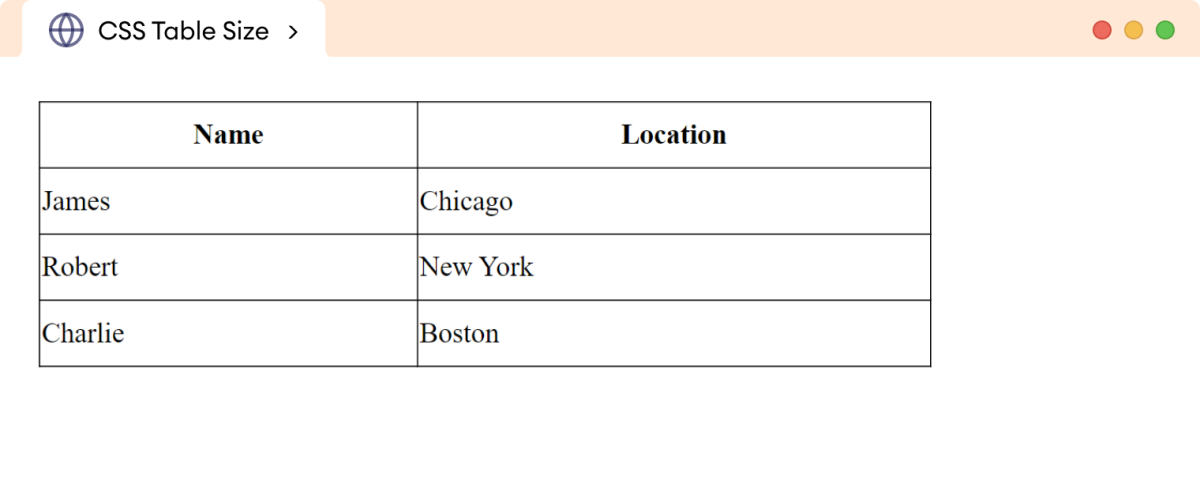
border-collapse: collapse;

border: 1px solid black;

width: 80%;

height: 170px;

}

**Browser Output**

**Table Layout**

The table-layout property specifies the structure and behavior of the table.

It can take one of the following values:

* auto: adjusts the column widths automatically based on content
* fixed: specifies the fixed column width

Let's see an example,

<table>

<tr>

<th>Header 1</th>

<th>Header 2</th>

<th>Header 3</th>

</tr>

<tr>

<td>

This table cell contains large amount of the text.

</td>

<td>Cell 2</td>

<td>Cell 3</td>

</tr>

</table>

Css:

/\* styles table with fixed layout \*/

table {

width: 500px;

table-layout: fixed;

}

table, th , td {

border: 1px solid black;

border-collapse: collapse;

}

Output:

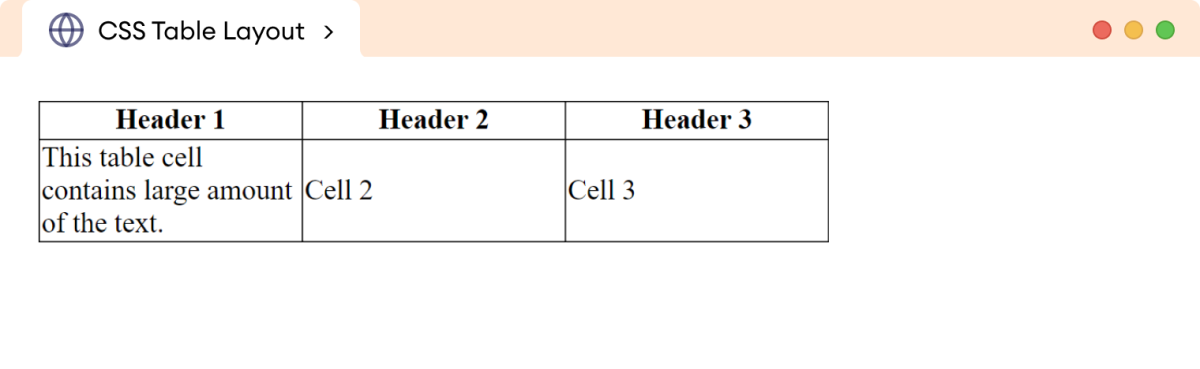


table-layout: auto;

**Horizontal Alignment**

The text-align property sets the horizontal alignment of the table content. For example

<body>

<h2>Horizontal alignment</h2>

<table>

<tr>

<th>text-align: left</th>

<th>text-align: right</th>

<th>text-align: center</th>

<th>text-align:justify</th>

</tr>

<tr>

<td class="first-data">Left</td>

<td class="second-data">Right</td>

<td class="third-data">Center</td>

<td class="fourth-data">Justify the content</td>

</tr>

</table>

</body>

Css

.first-data {

text-align: left;

}

.second-data {

text-align: right;

}

.third-data {

text-align: center;

}

.fourth-data {

text-align: center;

}

table {

width: 100%;

border-collapse: collapse;

}

table,th,td {

border: 1px solid black;

}

.first-data {

text-align: left;

}

.second-data {

text-align: right;

}

.third-data {

text-align: center;

}

.fourth-data {

text-align: center;

}

table {

width: 100%;

border-collapse: collapse;

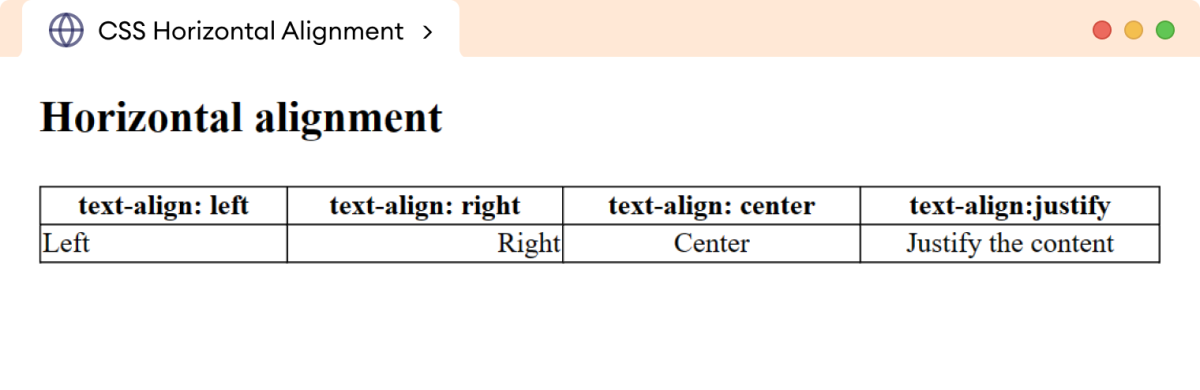
}

table,th,td {

border: 1px solid black;

}

Output:



**Vertical Alignment**

The vertical-align property sets the vertical alignment of the table content.

Let's see an example,

<h2>Vertical alignment</h2>

<table>

<tr>

<th>vertical-align: top</th>

<th>vertical-align: bottom</th>

<th>vertical-align: middle</th>

<th>vertical-align:baseline</th>

</tr>

<tr>

<td class="first-data">Top</td>

<td class="second-data">Bottom</td>

<td class="third-data">Middle</td>

<td class="fourth-data">Baseline</td>

</tr>

</table>

CSS:

.first-data {

vertical-align: top;

}

.second-data {

vertical-align: bottom;

}

.third-data {

vertical-align: middle;

}

.fourth-data {

vertical-align: baseline;

}

table {

width: 100%;

border-collapse: collapse;

}

table,th,td {

border: 1px solid black;

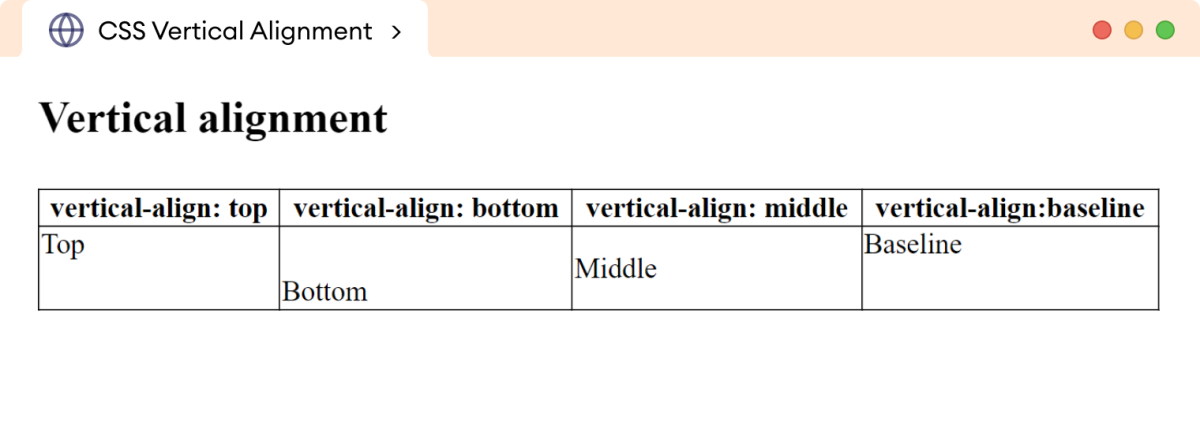
}

td {

height: 50px;

}

OUTPUT:



**Background Color**

The table can be styled with different background colors. For example,

<table>

<tr>

<th>Name</th>

<th>Location</th>

</tr>

<tr >

<td>James</td>

<td>Chicago</td>

</tr>

<tr >

<td>Robert</td>

<td>New York</td>

</tr>

</table>

CSS:

th {

background-color: greenyellow;

}

tr {

background-color: yellow;

}

table {

border-collapse: collapse;

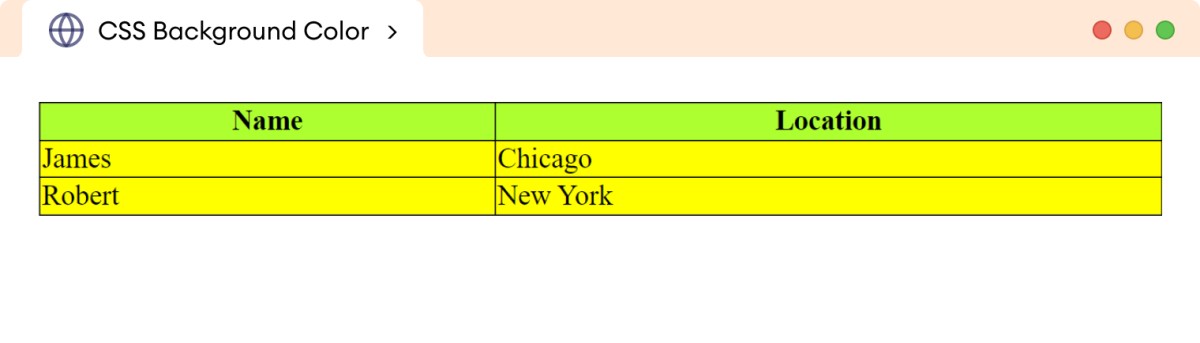
width: 100%;

}

table, th, td {

border: 1px solid black;

}



**Table Padding**

The padding property adds the space between the border and the content. For example,

<table>

<tr>

<th>Name</th>

<th>Location</th>

</tr>

<tr>

<td>James</td>

<td>Chicago</td>

</tr>

<tr>

<td>Robert</td>

<td>New York</td>

</tr>

<tr>

<td>Charlie</td>

<td>Boston</td>

</tr>

</table>

CSS:

th,td {

padding: 15px;

}

table {

border-collapse: collapse;

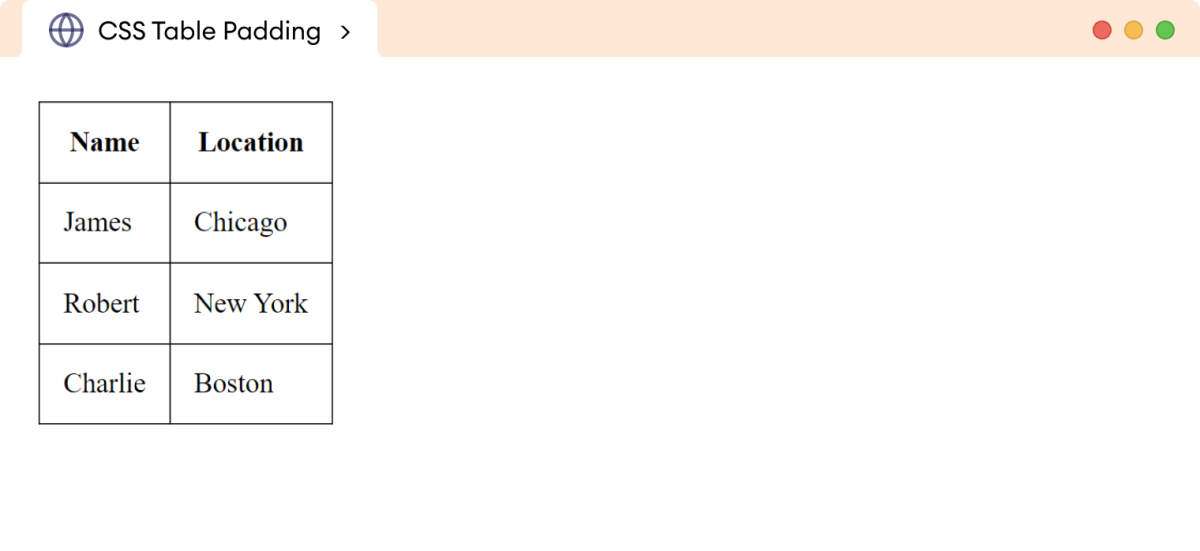
}

table,th,td {

border: 1px solid black;

}

OUTPUT:



In the above example,

padding: 15px

adds a 15 pixels gap between the border and the content in the table.

**Hoverable Table**

We can add the hover effect to the table elements with the help of the hover pseudo-class selector. For example,

Css:

tr:hover {

background-color: yellow;

}