CSS Div Layouts

Web Authoring and Design

Benjamin Kenwright

Outline

- Why use Div Layout instead of Tables?
- How do we use the Div Tag?
- How to create layouts using the CSS Div Tag?
- Summary
- Review/Discussion

Table vs Div Layouts

- **table**
  - ▷ Pros: supported by all browsers
  - ▷ Cons: bind style to content; hard to maintain
- **div**
  - ▷ Pros: easy to maintain
  - ▷ Cons: not supported by all browsers
- We recommend – **div**, reasons:
  - ▷ CSS is to separate **structure** from **content**.
  - ▷ Supporting most common/popular browsers are enough. May be it's time for some people to upgrade their browsers

What is Div Tag?

- CSS Division (div) is a container element and it is used to group related items together

- When ever there is a situation that you need to collect various objects into a larger container for scripting or styling purposes, div is the best solution

- The use of <div> tag is straightforward

Div Syntax

syntax :

    <div> ... </div>

e.g.

    <div>

        <p> A paagraph inside div </p>

    </div>

CSS Division

- CSS divisions to provide greater flexibility and mark out regions of the page.
- You can use divs by referencing the selector in the opening tag using ID and CLASS
  - ▷ e,g. id="myContainer" or class="myContainer"

Div in an HTML document

```
<html>
<head>
        <style type="text/css">
                #box {
                                width: 420px;
                                height:120;
                                border-width: 2px;
                                border-style:solid;
                                border-color:red;
                                background: #CCC;

                }
        </style>
</head>
<body>
        <div id="box">
                <h1>Box Model</h1>
                <p>
                The Box model determines how elements are positioned within the
                browser window. With the Box Model, a developer can control the
                dimensions, margins, padding, and borders of an HTML element.
                </p>
        </div>
</body>
</html>
```
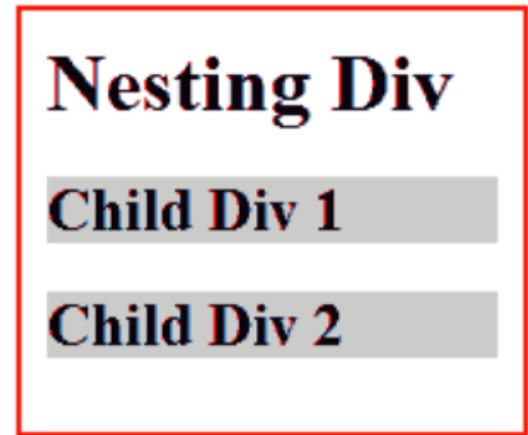
Output

Box Model

The Box model determines how elements are positioned within the browser window. With the Box Model, a developer can control the dimensions, margins, padding, and borders of an HTML element.

Nesting Div

■ The div element grouping a generic block of content that should be treated as a logical unit for scripting or styling purposes. A div can contain a number of other divs (child div) like HTML Tables. This is called Nesting Div

```
<div>
    <div>
        Child Div
    </div>
    <div>
        Child Div
    </div>
</div>
```

Nesting Div in an HTML page

```html
<html>
<head>
        <style type="text/css">
                    .parent {
                                        width: 200px;
                                        height:120;
                                        border-width: 2px;
                                        border-style:solid;
                                        border-color:red;
                                        padding:10px;

                    }
                    .child {

                                        overflow : hidden;
                                        background: #CCC;

                    }
        </style>
</head>
<body>
        <div class="parent">
                    <h1> Nesting Div</h1>
                    <div class="child">
                                        <h2>Child Div 1</h2>
                    </div>
                    <div class="child">
                                        <h2>Child Div 2</h2>
                    </div>
        </div>
</body>
</html>
```

Output

Careful

- Div elements very carefully and use only when it is necessary for logical structure or styling
- Excessive use of Div tags can make a page difficult to manage/debug/extend

Div Layouts

- Div tag to create your website layout
- Common website layouts

Common Layouts

- Variable width content:
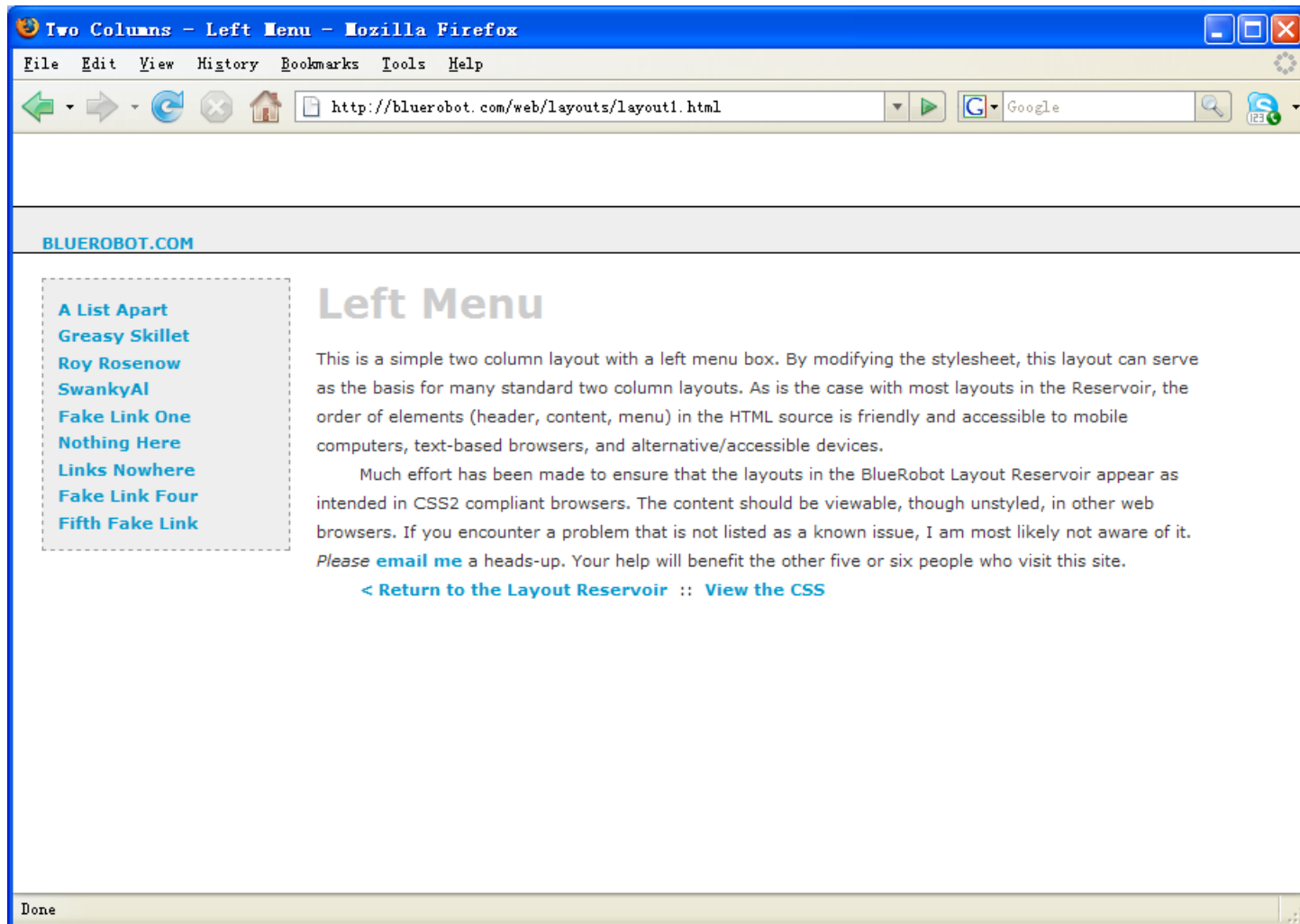 - ▷ 2 columns - left menu
  - ▷ 2 columns - right menu
  - ▷ 3 columns
- Centered (fixed width content):
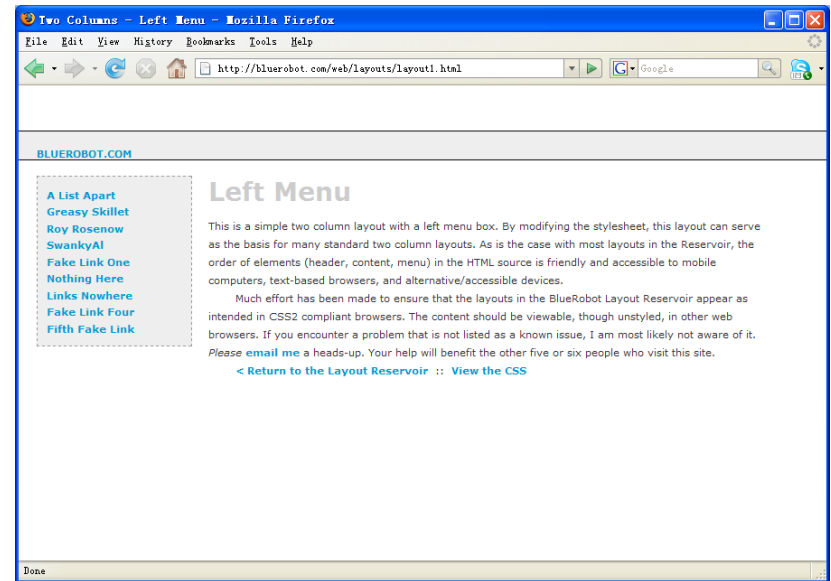 - ▷ 2 columns
  - ▷ 3 columns
- 4 columns (fluid/variable width)
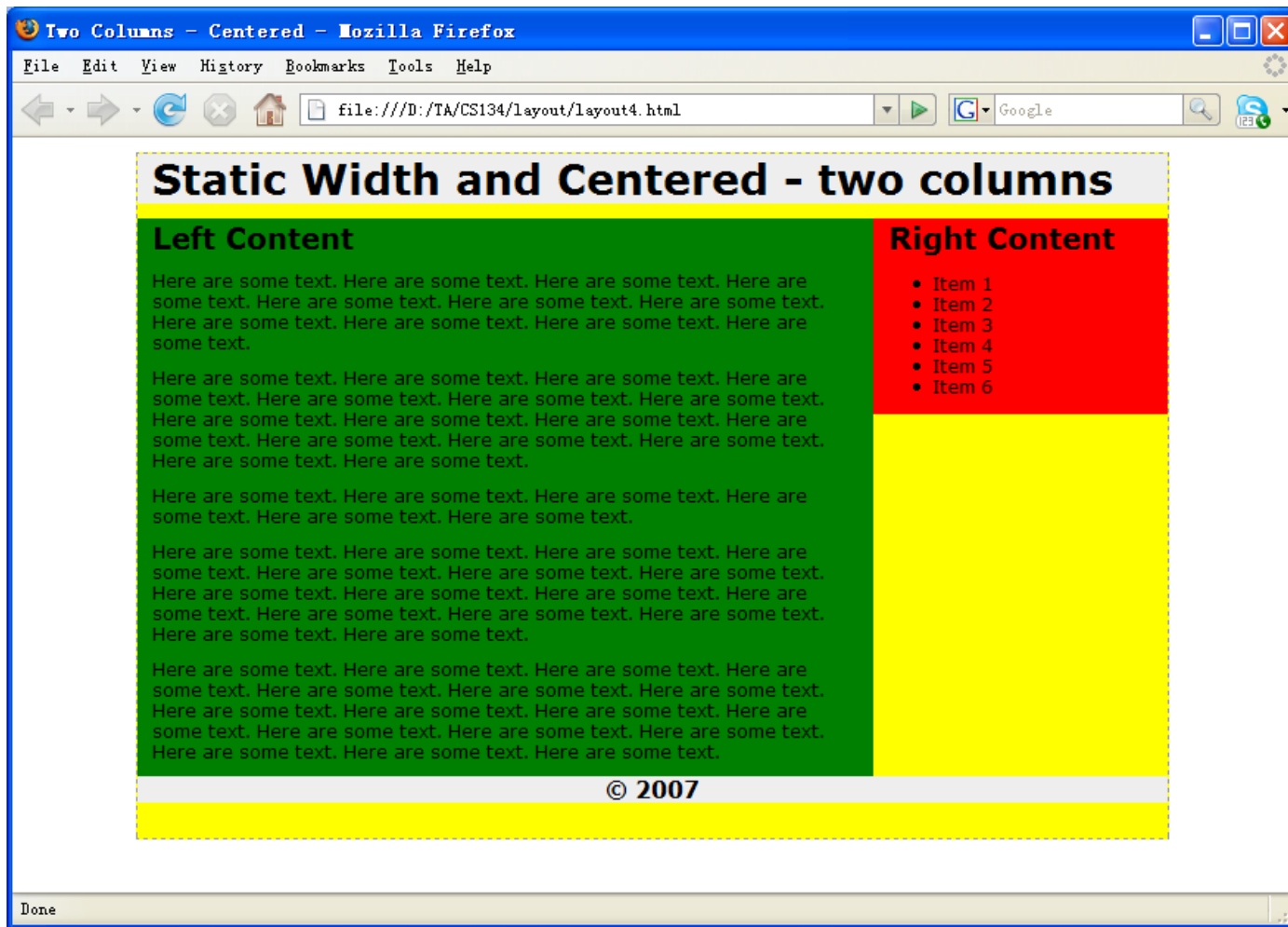- Many other …

Two Columns - Left Menu

Two Columns - Left Menu

```
#Header {
        margin:50px 0px 10px 0px;
        padding:17px 0px 0px 20px;
        border:1px dashed #999;
        background-color:#eee;
}
#Content {
        margin:0px 50px 50px 200px;
        padding:10px;
        border:1px dashed #999;
        background-color: #eee;
}
#Menu {
        position:absolute;
        top:100px;
        left:20px;
        width:150px;
        padding:10px;
        background-color:#eee;
        border:1px dashed #999;
}
```

Two Columns
Centered Fixed Width

Two Columns Centered Fixed Width

```
body {
        margin:0px;
        padding:0px;
        text-align: center;
}

#Wrapper {
        width:700px;
        margin-right:auto;
        margin-left:auto;
        border:1px dashed #999;
}

#Header {
        background: #eee;
}
```
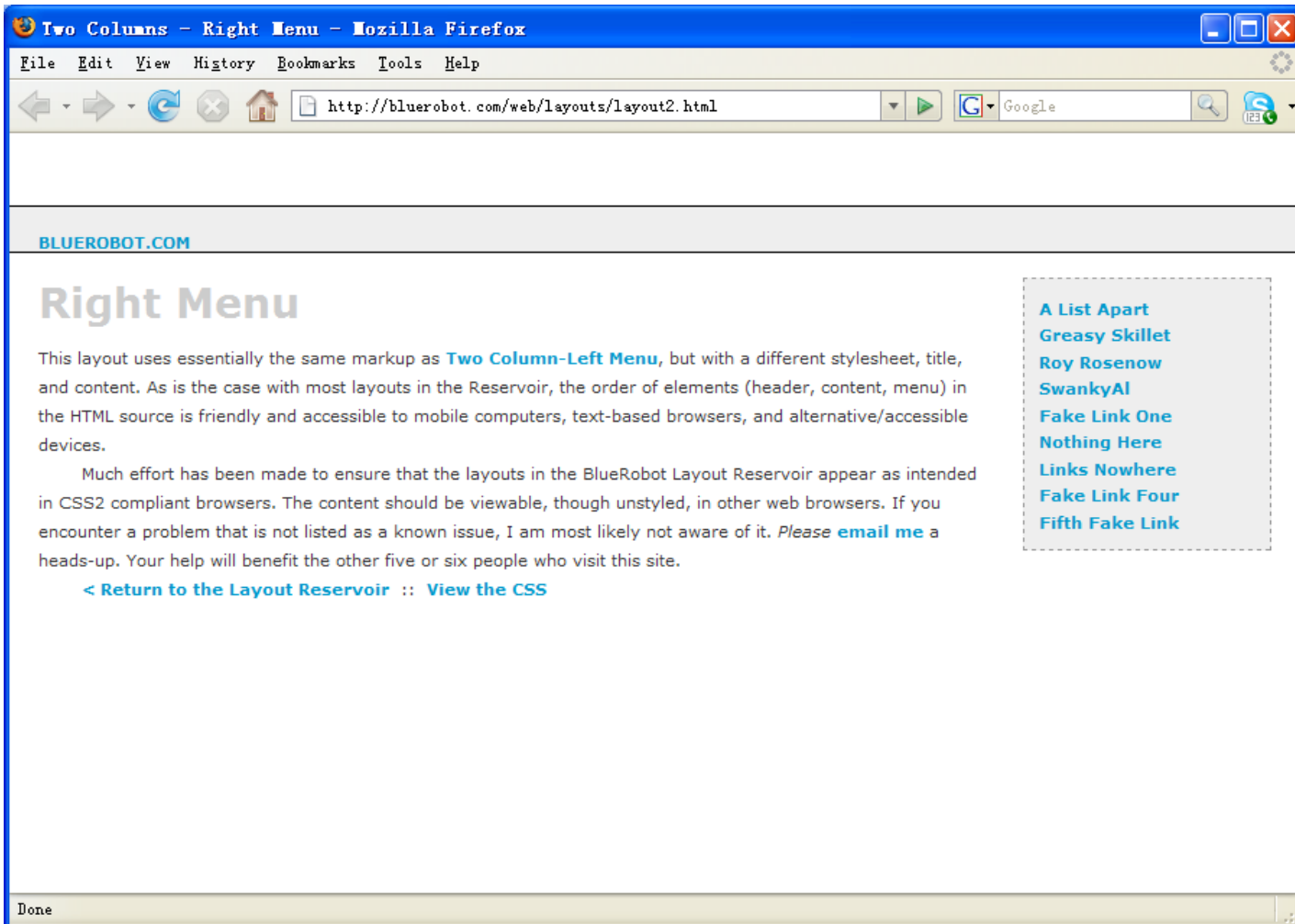
```
#Menu {
        float:right;
        width:200px;
        background: #eee;
}

#Content {
        float:left;
        width:500px;
        background: #666;
}

#Footer {
        clear: both;
        background: #eee;
}
```
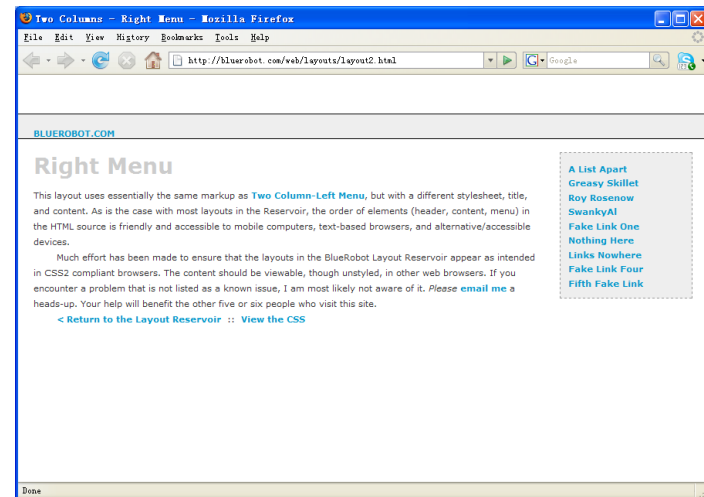
Two Columns - Right Menu
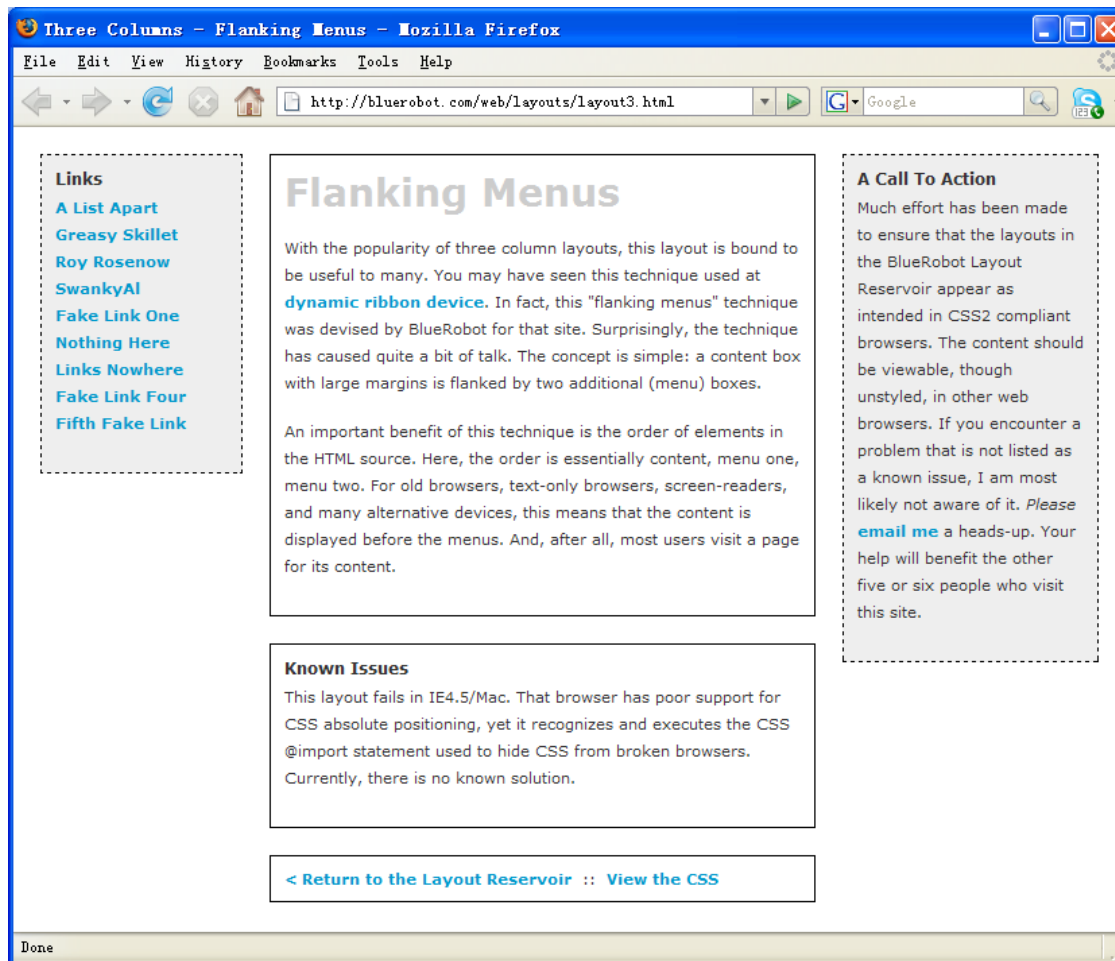
Two Columns - Right Menu

```
#Header {
        margin:50px 0px 10px 0px;
        padding:17px 0px 0px 20px;
        border:1px dashed #999;
        background-color:#eee;
}
#Content {
        margin:0px 200px 50px 50px;
        padding:10px;
        border:1px dashed #999;
        background-color: #eee;
}
#Menu {
        position:absolute;
        top:100px;
        right:20px;
        width:150px;
        padding:10px;
        background-color:#eee;
        border:1px dashed #999;
}
```

Three Columns Flanking Menu

Three Columns Flanking Menu

```
.content {
        position:relative;
        width:auto;
        min-width:120px;
        margin:0px 210px 20px 170px;
        border:1px solid black;
        padding:10px;
        z-index:3; /* This allows the content to
overlap the right menu in narrow windows in
good browsers. */
}
#navAlpha {
        position:absolute;
        width:128px;
        top:20px;
        left:20px;
        border:1px dashed black;
        background-color:#eee;
        padding:10px;
        z-index:2;
}
```

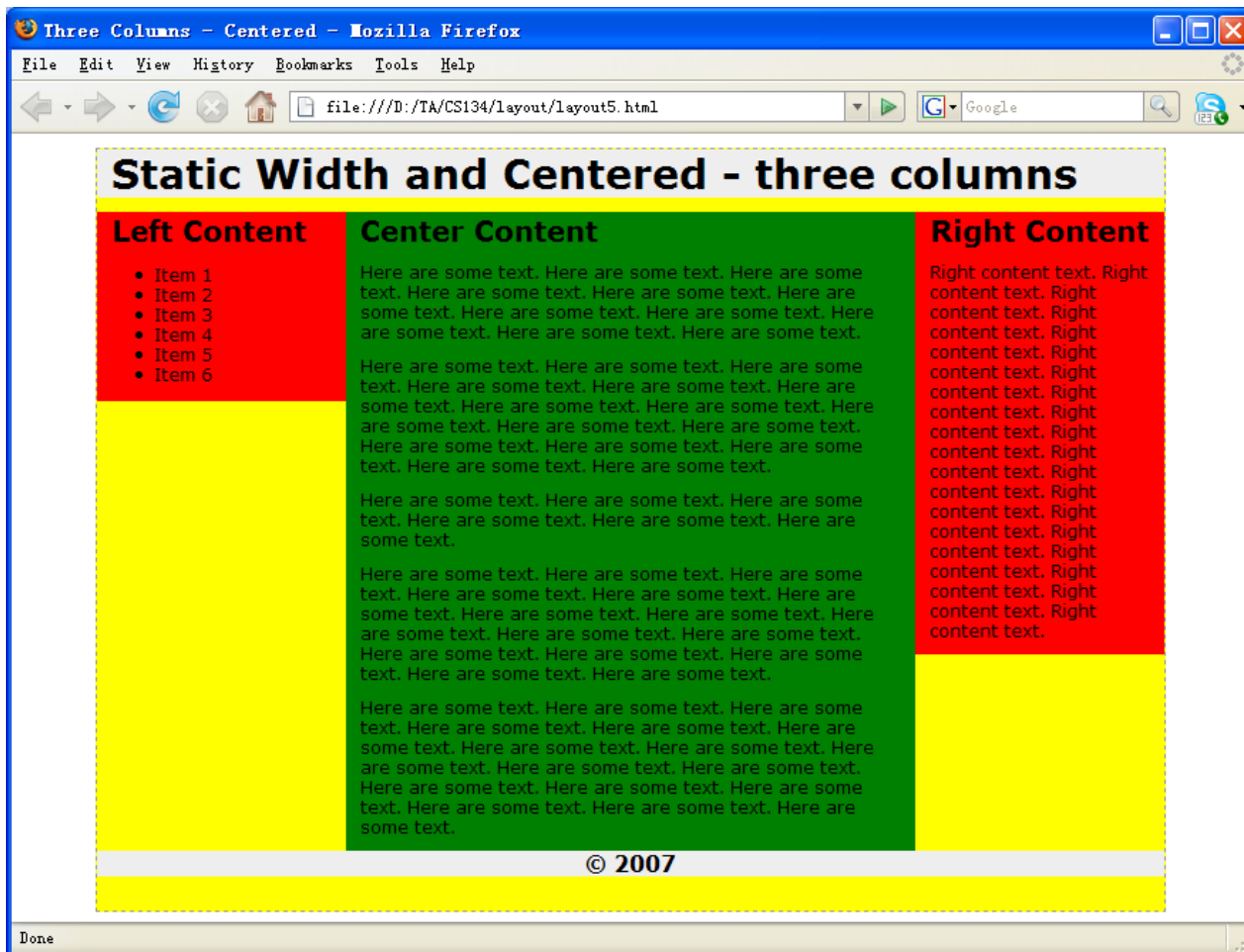```
#navBeta {
        position:absolute;
        width:168px;
        top:20px;
        right:20px;
        border:1px dashed black;
        background-color:#eee;
        padding:10px;
        z-index:1;
}
```

# Z-Index (or Stack Level)

- **z-index**: auto | <integer> | inherit
  - ▷ Z-axis positions are particularly relevant when boxes overlap visually.
  - ▷ In addition to their horizontal and vertical positions, boxes lie along a "z-axis" and are formatted one on top of the other.
  - ▷ Boxes with higher z-index stacked on top of the boxes with lower z-index.
  - ▷ Boxes with the z-index are stacked back-to-front according to document tree order.

# Three Columns Centered Fixed Width

# Three Columns Centered Fixed Width

```
body {
        text-align:center;
        margin:0px;
        padding:0px;
        font:12px verdana, arial, helvetica,
sans-serif;
}

#frame {
        width:750px;
        margin-right:auto;
        margin-left:auto;
        margin-top:10px;
        text-align:left;
        border:1px dashed #999;
        background-color: yellow;
}

#topcontent {
        background-color: #eee;
}
```
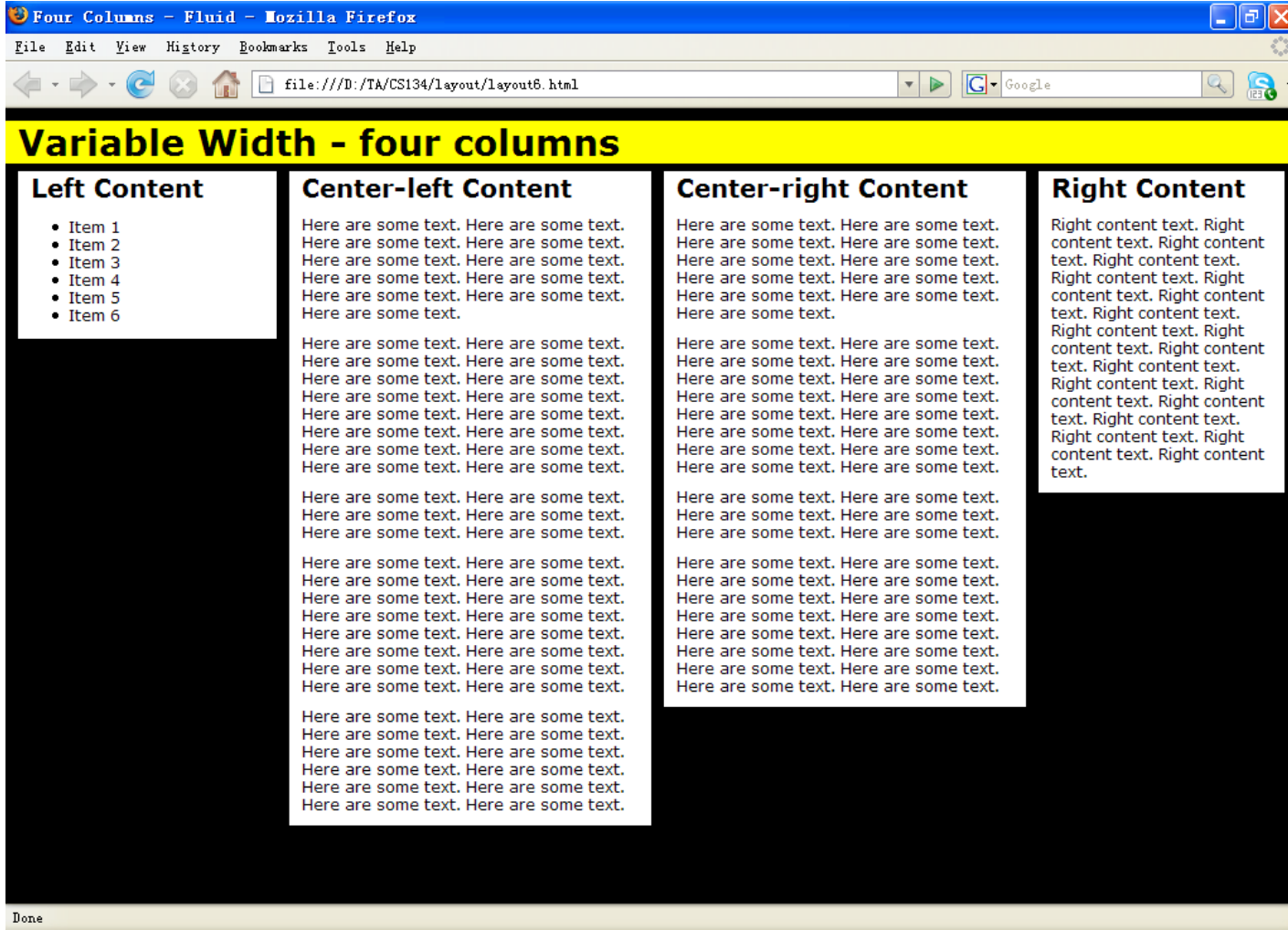
```
#centercontent {
        float:left;
        width:400px;
        background-color: green;
}
#leftcontent {
        float:left;
        width:175px;
        background-color: red;
}
#rightcontent {
        float:left;
        width:175px;
        background-color: red;
}

#bottomcontent {
        background-color:#eee;
        text-align:center;
}
```

# Four Columns Variable Width

# Four Columns Variable Width

```
#topcontent {
        background-color: yellow;
}

#leftcontent {
        position: absolute;
        left:1%;
        width:20%;
        top:50px;
        background:#fff;
}

#centerleft {
        position: absolute;
        left:22%;
        width:28%;
        top:50px;
        background:#fff;
}
```

```
#centerright {
        position: absolute;
        left:51%;
        width:28%;
        top:50px;
        background:#fff;
}

#rightcontent {
        position: absolute;
        left:80%;
        width:19%;
        top:50px;
        background:#fff;
}
```

CSS Table

- HTML table styling with CSS properties
- CSS table properties offer better control of the presentational aspects of the Table

Example

■CSS

```
table
{
        width:30%;
        background-image: url(bgpic.png);
        box-shadow: 10px -10px 5px #CCC;
}
```

■Output

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green
1004	Ford	Yellow

Table width and height in CSS

- Table width and height, use CSS width, height properties

- For example, table width as 30% and height of the td set to 40px

```
table
{
        width:40%;
}
td
{
        height:40px;
}
```

Output:

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green

```html
<!DOCTYPE html>
<html>
<head>
<style>
    table
    {
        width:30%;
    }
    td
    {
        height:40px;
    }
</style>
</head>
<body>
    <table border=1>
        <tr>
        <th>Roll No</th>
        <th>Name</th>
        <th>Team</th>
        </tr>
        <tr>
        <td>1001</td>
        <td>John</td>
        <td>Red</td>
        </tr>
        <tr>
        <td>1002</td>
        <td>Peter</td>
        <td>Blue</td>
        </tr>
        <tr>
        <td>1003</td>
        <td>Henry</td>
        <td>Green</td>
        </tr>
    </table>
</body>
</html>
```

Table column width in CSS

■ Specify column width in CSS, use the width property to td

Output

```
td
{
    width: 170px;
}
```

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green

CSS Table Row height

■ Set Row Height through CSS line-height property it set to each tr.

```
tr
{
    line-height: 50px;
}
```

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green

CSS Table border

- Table border in CSS, use the CSS border property

```
table,th,td
{
    border:2px solid green;
}
```

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green

```html
<!DOCTYPE html>
<html>
<head>
<style>
    table,th,td
    {
        border:2px solid green;
    }
    th, td
    {
        width:100px;
        height:40px;
    }
</style>
</head>

<body>
    <table>
        <tr>
        <th>Roll No</th>
        <th>Name</th>
        <th>Team</th>
        </tr>
        <tr>
        <td>1001</td>
        <td>John</td>
        <td>Red</td>
        </tr>
        <tr>
        <td>1002</td>
        <td>Peter</td>
        <td>Blue</td>
        </tr>
        <tr>
        <td>1003</td>
        <td>Henry</td>
        <td>Green</td>
        </tr>
    </table>
</body>
</html>
```

CSS Collapse Table borders

- CSS Collapse property takes two values, separate and collapse

- separate : The separate value forced all cells have their own independent borders and allow spaces between those cells.

- collapse : This value collapse all spaces between table borders and cells, so you can see as a single line border

```
table
{
    border-collapse:collapse;
}
```

Output:

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green

CSS Table Text Align

■Align text horizontally and vertically in CSS.

▷Horizontal: text-align property

▷Vertically: vertical-align property

Property	Value
Text-align	Right | left | center | justify
Default : left	

Property	Value
vertical-align	baseline | sub | super | top | text-top | middle | bottom | text-bottom
Default : baseline	

```
td
{
   text-align:right;
   vertical-align:bottom;
}
```

CSS Table Cell Padding

■ CellPadding is used to control the space between the contents of a Cell and the Cell borders

```
td
{
    padding: 10px;
}
th
{
    padding: 20px;
}
```

Output:

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green

CSS Table Cell Spacing

- Cellspacing attribute places space around each cell in the table. To specify cell Spacing in CSS, use the CSS border-spacing property

```
table,th,td
{
   border:2px solid green;
   border-spacing: 20px;
}
```

Output:

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green

```html
<!DOCTYPE html>
<html>
<head>
<style>
    table,th,td
    {
        border:2px solid green;
        border-spacing: 20px;
    }
</style>
</head>

<body>
    <table>
        <tr>
        <th>Roll No</th>
        <th>Name</th>
        <th>Team</th>
        </tr>
        <tr>
        <td>1001</td>
        <td>John</td>
        <td>Red</td>
        </tr>
        <tr>
        <td>1002</td>
        <td>Peter</td>
        <td>Blue</td>
        </tr>
        <tr>
        <td>1003</td>
        <td>Henry</td>
        <td>Green</td>
        </tr>
    </table>
</body>
</html>
```

CSS Table background Image

```
table
{
    background-image: url(your image file);
}
```

Output:

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green
1004	Ford	Yellow

CSS Shadow on a Table

```
table
{
  box-shadow: 10px -10px 5px #CCC;
}
```

Output:

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green
1004	Ford	Yellow

CSS Rounded Corners Table

```
table
{
    border-radius:25px
}
```

Output:

Highlight CSS Table Row on Hover

```css
tr:hover
{
  background-color: #ffff99;
}
```

Move mouse over the rows

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green
1004	Ford	Yellow

This Week

- Review Slides

- Read Associated Chapters

- Weekly Tasks
  - ▷ Update GitHub Account/Webpage

- **Online Quizzes**

Summary

- Overview of CSS Div Layouts

- Tables and CSS Styles

- Hands-On/Practical

Questions/Discussion