

Chapter 5-Frames

Frames:

- You may have visited web sites in which you were able to scroll the main portion of the web page while a smaller section containing navigation remained stationary on the screen. Although this looked as though it were all contained on a single web page, actually multiple web pages appeared on the screen at the same time, and each was displayed in a frame.
- Frames are created using HTML, but you can interact and manipulate frames using a JavaScript. You'll see how this is done in this chapter.
- All frames contain at least three web pages. The first frame surrounds the other frames, and this entire collection is called the *frameset*. The other frames are within the frameset, and each is referred to as a *child*. You can give each child a unique name so you can later refer to it in your application.
- JavaScript refers to the frameset as the *top* or the *parent*. The parent frame is always at the top of the display. Child windows appear within the parent window. You can nest frames many layers deep—so the top level may actually still be a child frame of another frameset.

Example1:

```
<html>
<head>
<title>Create a Frame</title>
</head>

<frameset rows="50%,50%">

<frame src="WebPage1.html" name="topPage"/>
<frame src="WebPage2.html" name="bottomPage"/>

</frameset>

</html>
```

WebPage1.html

```
<html>
<head>
<title>Web Page 1</title>
</head>
<body>
<FORM action="http://abc.com" method="post">

<P>

<INPUT name="WebPage1" value="Web Page 1"
type="button" />

</P>
</FORM>
</body>
</html>
```

WebPage2.html

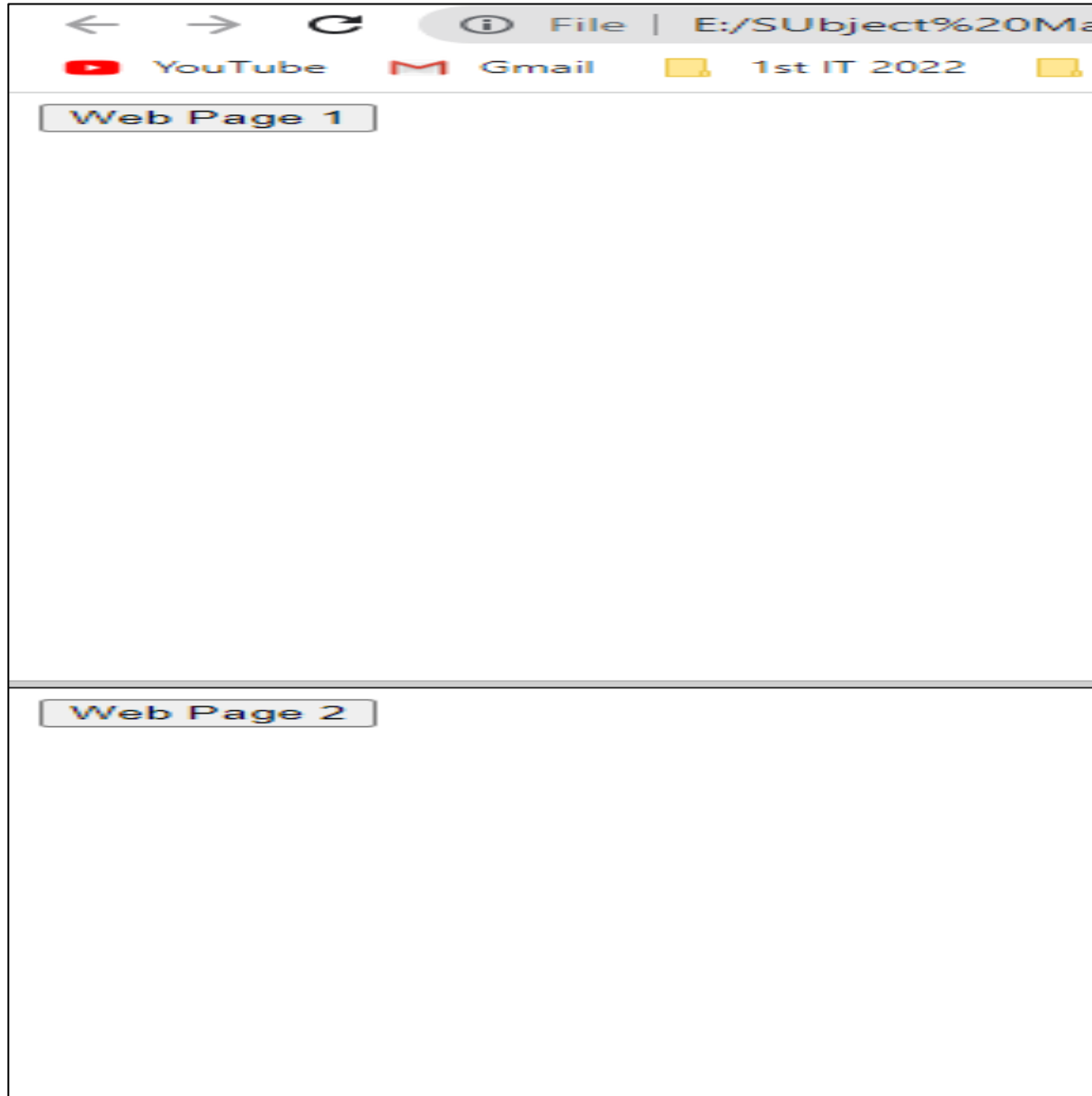
```
<html>
<head>
<title>Web Page 2</title>
</head>
<body>
<FORM action="http://xyz.com" method="post">

<P>

<INPUT name="WebPage2" value="Web Page 2"
type="button" />

</P>
</FORM>
</body>
</html>
```

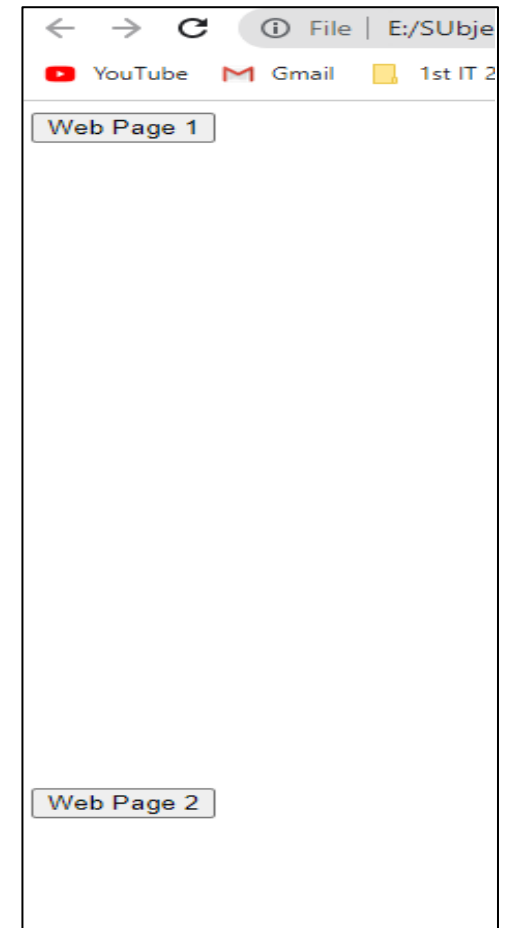
OUTPUT



Invisible Borders

- You can make it less obvious that you are using frames by hiding the borders around the child windows within your frameset. The result appears as one web page on the screen, even though in reality each of multiple web pages appears in its own child window.
- The border can be hidden by setting the frameborder and border attributes of the <frame> tag to zero (0).

```
<html>
<head>
<title>Create a Frame</title>
</head>
<frameset rows="50%,50%">
<frame src="WebPage1.html" name="topPage" frameborder="0" border="0" />
<frame src="WebPage2.html" name="bottomPage" frameborder="0" border="0" />
</frameset>
</html>
```



- [border](#): This attribute of frameset tag defines the width of the border of each frame in pixels. Zero value is used for no border.
- [frameborder](#): This attribute of frameset tag is used to specify whether a three-dimensional border should be displayed between the frames or not for this use two values 0 and 1, where 0 defines no border and value 1 signifies for yes there will be a border.

Calling a Child Window's JavaScript Function

WebPage1.html

```
<html>
<head>
<title>Web Page 1</title>
<script>
<!--
function ChangeContent()
{
alert("Function Called");
}
-->
</script>

</head>
<body>
<FORM action="http://www.jimkeogh.com"
method="post">
<P>
<INPUT name="WebPage1" value="Web
Page 1" type="button" />
</P>
</FORM>
</body>
</html>
```

WebPage2.html

```
<html>
<head>
<title>Web Page 2</title>
</head>
<body>
<FORM action="http://www.jimkeogh.com"
method="post">
<P>

<INPUT name="WebPage2"
value="Web Page 2"
type="button"
onclick="parent.topPage.ChangeContent()"
/>

</P>
</FORM>
</body>
</html>
```

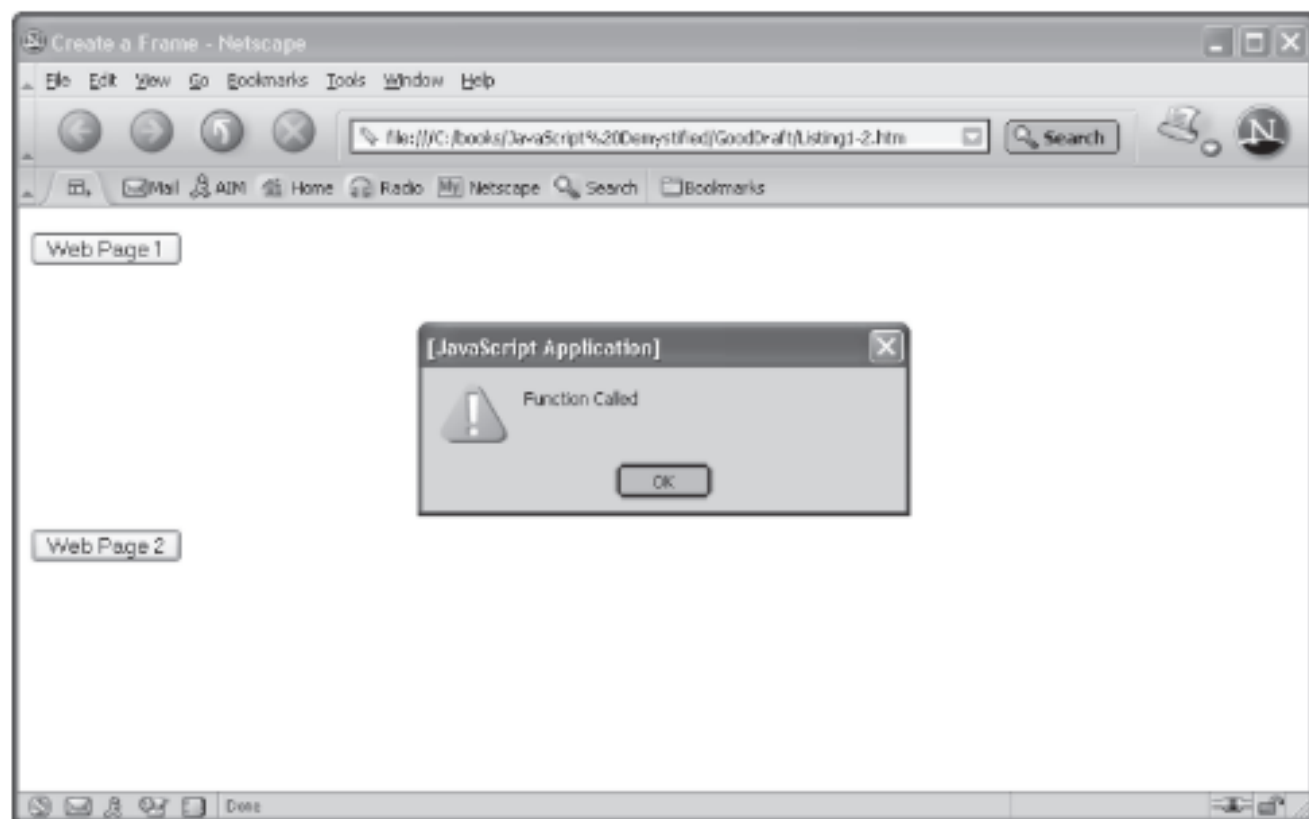
Frame.html

```
<html>
<head>
<title>Create a Frame</title>
</head>
<frameset rows="50%,50%">

<frame src="WebPage1.html"
name="topPage"/>



<frame src="WebPage2.html"
name="bottomPage"/>

</frameset>
</html>
```

Web Page 1

Web Page 2

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder  Performance insights 

  top   Filter

✖ ▶ Uncaught DOMException: Blocked a frame with origin "null" from accessing a cross-origin frame.
at HTMLInputElement.onclick (<file:///E:/SUBject%20Material%20yearwise/jan%202023-june%202023/FSL/Programs/WebPage22.html#11:73>)

>

Changing the Content of a Child Window

You can change the content of a child window from a JavaScript function by modifying the source web page for the child window. To do this, you must assign the new source to the child window's href attribute.

WebPage1.html

```
<html>
<head>
<title>Web Page 1</title>
<script language="Javascript" type="text/javascript">
<!--
function ChangeContent()
{
parent.topPage.location.href='WebPage3.html'
}
-->
</script>
</head>
<body>
<FORM action="http://www.jimkeogh.com"
method="post">
<P>
<INPUT name="WebPage1" value="Web Page 1"
type="button" onclick="ChangeContent()"/>
</P>
</FORM>
</body></html>
```

WebPage2.html

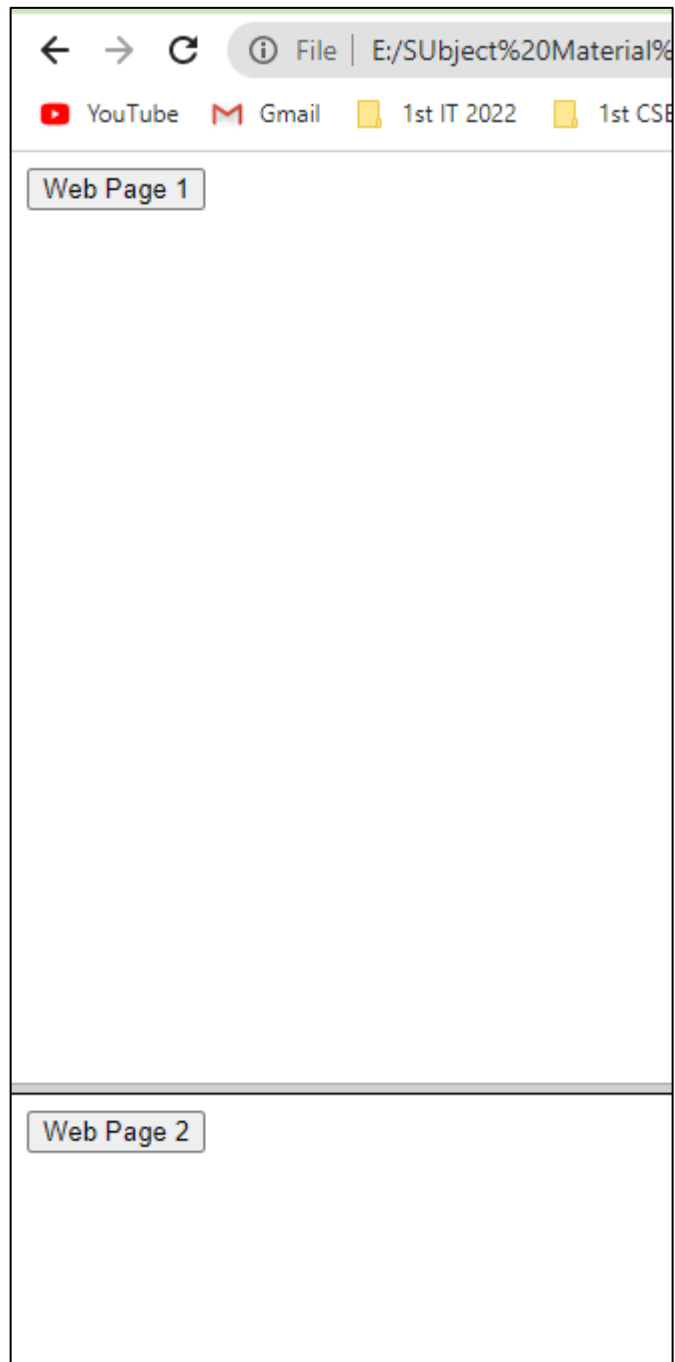
```
<html>
<head>
<title>Web Page 2</title>
</head>
<body>
<FORM
action="http://www.jimkeogh.co
m" method="post">
<P>
<INPUT name="WebPage2"
value="Web Page 2"
type="button" />
</P>
</FORM>
</body>
</html>
```

WebPage3.html

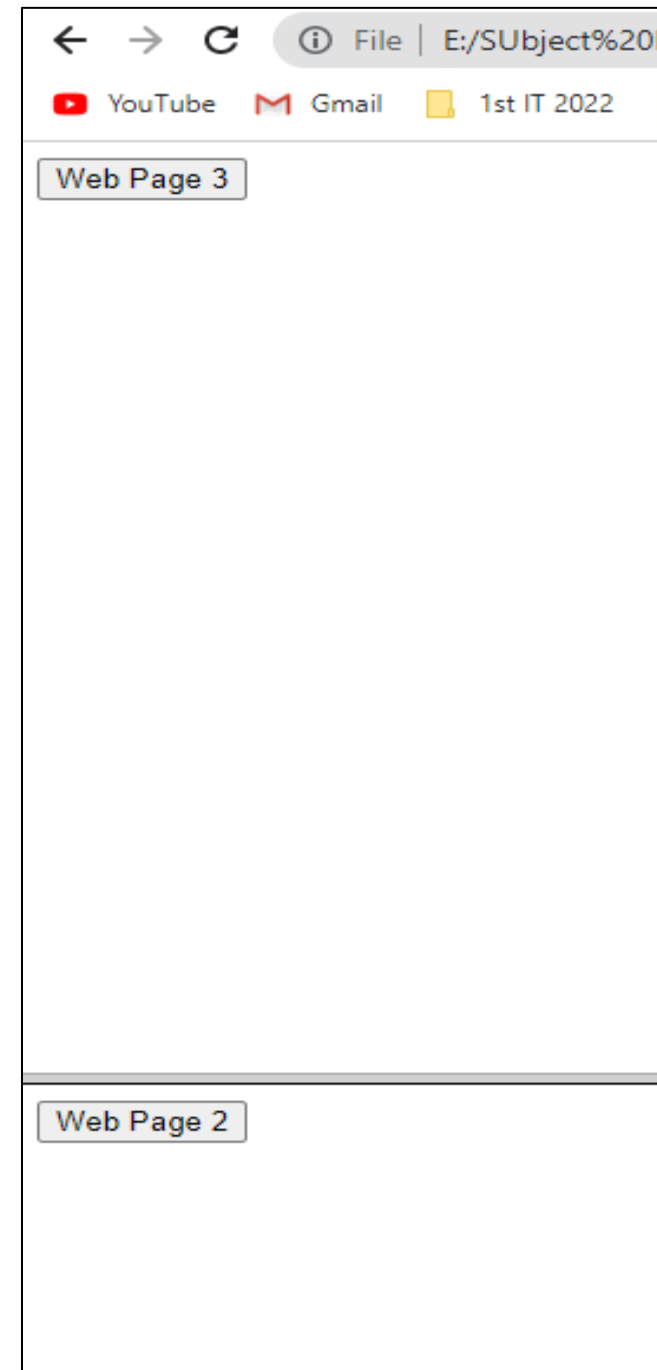
```
<html>
<head>
<title>Web Page 3</title>
</head>
<body>
<FORM
action="http://www.jimkeog
h.com" method="post">
<P>

<INPUT name="WebPage3"
value="Web Page 3"
type="button" />

</P>
</FORM>
</body>
</html>
```



Click on [Web Page1](#)



Writing to a Child Window from a JavaScript

Typically, the content of a child window is a web page that exists on the web server. However, you can dynamically create the content when you define the frameset by directly writing to the child window from a JavaScript. The JavaScript must be de-fined in the HTML file that defines the frameset and called when the frameset is loaded.

```
<html><head>
<title>Create a Frame</title>
<script language="Javascript" type="text/javascript">
<!--
function ChangeContent()
{
window.topPage.document.open()
window.topPage.document.writeln(
'<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN">')
window.topPage.document.writeln(
'<html>')
window.topPage.document.writeln('<head>')
window.topPage.document.writeln(
'<title>Web Page 3</title>')
window.topPage.document.writeln('</head>')
window.topPage.document.writeln('<body>')
window.topPage.document.writeln(
'<FORM action="http://www.jimkeogh.com" method="post">')
window.topPage.document.writeln('<P>')
```

----- The writeln() method writes directly to an open (HTML) document stream.

```
window.topPage.document.writeln(
'<INPUT name="WebPage3" value="Web Page 3"
type="button" />')
window.topPage.document.writeln('</P>')
window.topPage.document.writeln('</FORM>')
window.topPage.document.writeln('</body>')
window.topPage.document.writeln('</html>')
window.topPage.document.close()
}
-->
</script>
</head>
<frameset rows="50%,50%" onload="ChangeContent()">
<frame src="WebPage1.html" name="topPage" />
<frame src="WebPage2.html" name="bottomPage" />
</frameset>
</html>
```

- To write dynamic content to a child window, you must assign a source file to each frame of the frameset, even though you are dynamically creating the source for at least one of those frames. You'll notice in this example that WebPage1.html is assigned to the topPage frame. WebPage1.html must be a real file, although it won't appear in the topPage frame because the JavaScript function writes the content to that frame.
- The JavaScript function is defined in the <head> tag and is called when the onload event occurs. The topPage child window must be opened before the JavaScript function can write to the window. You open the child window by calling the open() method for that frame, as shown here:

```
window.topPage.document.open()
```

- Once opened, call the write() method to write HTML content to the child window to create the web page. This example displays the Web Page 3 button on a form. The final step is to call the close() method to close the window, as shown here:

```
window.topPage.document.close()
```