

Chapter 4. HTML Tables

Table of Contents

Objectives	1
4.1 Introduction to Tables	1
4.1.1 What is a Table?	1
4.1.2 Why do We Use Tables?	2
4.1.3 Creating a Data Table	2
4.1.4 Activity 2: HTML Colour Table.....	6
4.2 Using Tables in Page Design	6
4.2.1 Using Tables in Page Design	6
4.2.2 Flexible Design.....	7
4.2.3 Fixed Design.....	7
4.2.4 Activity 3: Using rowspan	8
4.2.5 Activity 4: Using colspan, cellspacing and cellpadding	9
4.2.6 Activity 5: More cellspacing and cellpadding	12
4.2.7 Activity 6: Fixed and flexible Web page design.....	12
4.2.8 Activity 7: Time Table.....	13
4.3 Review Questions	13
4.4 Discussions and Answers.....	13
4.4.1 Correct code for Activity 1 step 1.....	13
4.4.2 Solution to Activity 1 step 5	14
4.4.3 Solution to Activity 2: HTML Color Table	14
4.4.4 Discussion Topic	15
4.4.5 Answers to Review Questions	16

Objectives

At the end of this unit you will be able to:

- explain why tables are used to organize and display data;
- construct a table using HTML5 tags;
- insert data into the relevant table cells;
- add colour to particular table cells;
- discuss the advantages and disadvantages of fixed and flexible Web page design;
- implement a table in a flexible Web page using relative measurements;
- implement a table in a fixed Web page using pixel measurements.

4.1 Introduction to Tables

4.1.1 What is a Table?

A table is a grid organized into columns and rows, much like a spreadsheet. An example table is shown below. This table consists of sixteen cells organized into rows and columns. But before beginning to use tables in website design, we should consider the role that they fill. Working with tables in HTML5 has become more powerful due to the new HTML5 table tags and other elements available in HTML5.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

4.1.2 Why do We Use Tables?

Tables were initially developed as a method to organize and display data in columns and rows. This chapter discusses such tables. However, tables later became a tool for Web page layout, and as such provide a possible solution for structured navigation. Frames may also be used to provide structured navigation. However, the use of tables over frames is preferred for this purpose, as earlier Web browsers (e.g. Netscape ver.1.0) do not support frames.

To Do

Read up about tables in your textbooks.

4.1.3 Creating a Data Table

Work through Activity 1 in order to understand how tables are created. Bear in mind that rarely is anything achieved which satisfies all of the stated requirements in the first pass. The key to developing perfect Web pages relies on that old adage: "*Learn from your mistakes!*"

Therefore, as long as a start is made, and mistakes are seen as a learning experience, then the design process will eventually succeed.

Please feel free to experiment at any time. If you make mistakes but manage to correct them, take encouragement from this.

Activity 1: Creating a Table

The objective of this Activity is to create a timetable for CSC5003 students to be displayed on a Web page as shown below:

CSC503 timetable

	Monday	Tuesday	Wednesday	Thursday	Friday
6-7pm	look at website	free	Implementation	free	free
7-8pm	take some notes	free	Implementation	free	free

1. Begin a new Web page in your text editor. The header is shown below. When entering the text, try to spot the deliberate mistake and correct it as necessary.

```
<HTML>
  <HEAD>
    <TITLE>
      HTML Table Design
    </HEAD>
  </TITLE>
  <BODY>
  </BODY >
</HTML >
```

The correct code is given at the chapter's end.

2. Save your file as tab_ex1.html
3. The next stage is to open the table. To open and close a table, use respectively the <TABLE> and </TABLE> tags within the document's BODY.

```

<HTML>
<HEAD>
<TITLE>
  HTML Table
  Design
</HEAD>
</TITLE>
<BODY>
<TABLE>
</TABLE>
</BODY >
</HTML >

```

4. Save the file and load it in your browser. At first you will notice a blank screen as the table is not visible. A border and rows may be added to make the table visible. If you do not specify a border for the table, it will be displayed without borders. When adding a border, its size can be defined in pixels, for example: `<TABLE border=10 style= "width: 80%" >`. Notice the use of the width attribute to set the table to a width of 80% of the screen's size (this can also be defined in pixels). However, it is worth noting that the border attribute is on its way out of the HTML standard! It is better to use CSS by first creating a `<style>` tag within the `<head>` tag then leave using only the style attribute within the table tag. 'td' stands for 'tabular data' and 'th' stands for 'tabular header'.

```

<style>
table, th, td {
  border: 1px
solid black;
}
</style>
....
<TABLE style=
"width: 80%" >

```

5. The `<TR>` tag is used to add rows. Each row is composed of several data cells. Their dimensions can be defined using width and height attributes: `<TD width=25% height=20 bgcolor="darkred">` Notice that the cell's colour can also be defined. Try to create the table below before you look at the solution code under Discussion and Answers at the end of the chapter.

red cell	light blue cell
----------	-----------------

Figure 5.1: Table with one row and two columns

6. Reopen the file `tab_ex1.html` in your text editor and make the following amendments to `<TABLE>` and `<tr>` tags. Note the `<CENTER>` tag centers the table horizontally and it also centers the text within each cell in the row.

```

<TABLE style= "width: 80%" align = "center">

<tr align = "center">

```

7. Save this file as `tab_ex2.html` and view it in your browser. It should look as below.

red cell	light blue cell
----------	-----------------

Figure 5.2: Table with centered text

8. We can see that the text is still not given any specific font. HTML `` tag is deprecated in version 4.0, onwards (hence it is not supported in HTML5) and now all fonts are set by using CSS. Try to assign the Comic Sans MS font by making the following addition to the style section. Save the file as `tab_ex4.html`.

```
font-family: Comic Sans MS;
```

This sets all the text in in each cell to have the same font. What if you want to have different fonts in each cell? To do this, you can use the `<p style>` tag within each `<TD>` tag. Modify your `<TD>` tags to the following:

```
<TD width=25% height=70 bgcolor="red"> <p style="font-family: verdana">red  
cell </p> </td>  
<TD width=75% bgcolor="lightblue"> <p style="font-family: Comic Sans MS">  
light blue cell </p></td>
```

9. To add a caption to a table use the `<caption>` tag within the `<table>` body. This caption appears on top of the table. Add the caption “Tabling” to your table thus:

```
<caption> Tabling </caption>
```

10. Save the file as `tab_ex3.html` and view it in your browser. It should look as below.

Tabling

red cell	light blue cell
----------	-----------------

Figure 5.3: Table with caption and text with different font

11. In order to meet the objective of this Activity — that is, to create a timetable for CSC5003 — use the HTML code in the next page. Save this as `tab_ex4.html`. One extra HTML tag needs to be introduced: the `TH` tag, which inserts a table header cell. It is similar to the `TD` tag and has the same attributes (i.e. `align`, `bgcolor`, `height` etc.). However, `TH` is used to set the cell's text apart from the rest of the table's text, usually setting it bold and slightly larger. Now that you have completed Activity 1, you should have a good idea of how to create a basic data table.

```

<HTML>
<HEAD>
<TITLE>
    HTML Table Design
</TITLE>

<style>
table, th, td {
    border: 1px solid black;
}
</style>

</HEAD>
<BODY>

    <TABLE style= "width: 80%" align = "center">
    <caption> CSC503 timetable </caption>
    <tr >
        <td width=50%> </td>
        <th width = 150> Monday </th>
        <th width = 150> Tuesday</th>
        <th width = 150> Wednesday </th>
        <th width = 150> Thursday</th>
        <th width = 150> Friday</th>
    </tr>
    <tr >
        <td > 6-7pm </td>
        <td > Look at website</td>
        <td > free </td>
        <td > Implementation </td>
        <td > free </td>
        <td > free </td>
    </tr>
    <tr >
        <td > 7-8pm </td>
        <td > Take some notes</td>
        <td > free </td>
        <td > Implementation </td>
        <td > free </td>
        <td > free </td>
    </tr>
    </TABLE>

</BODY >
</HTML >

```

Here are instructions on how to organise and display data in a table:

1. Insert the <TABLE> tag and decide on the table's dimensions (if required)
2. Add a row using the <TR> tag
3. In the newly created row, insert a cell <TD> with the necessary dimensions and other attributes
4. Add the data to be displayed
5. Terminate the data cell </TD>
6. Repeat steps 3-5 as necessary
7. Terminate the row </TR>
8. Repeat steps 2-7 until all the necessary rows have been added
9. Terminate the table </TABLE>

To Do

Look up the basic table structure in your textbooks and on the Internet. Draw up a list of the tags for your own use and reference.

Check your list against this one:

HTML tag	Comments
<TABLE> </TABLE>	Table definition and end tag
<CAPTION> </CAPTION>	Caption definition and end tag
<TR> </TR>	Row definition and end tag
<TD> </TD>	Cell definition and end tag

4.1.4 Activity 2: HTML Colour Table

This Activity's objective is to write the HTML code to display the following table. Feel free to add more colours.

Some HTML Colors

Colour	Name	hexidecimal	RGB value
	Salmon	FA8072	250-128-114
	Gold	FFD700	255-215-0

See the end of the chapter for the solution.

To Do

Read up on 'Spanning Rows and Columns' and 'Table Appearance and Colours' in your textbooks. Add the new tags to your list of table related tags.

4.2 Using Tables in Page Design

4.2.1 Using Tables in Page Design

Tables are useful for laying out text and images on in Web page. Before continuing with instructions on how to do this, let us first consider why there is a need to manage layout.

It is important to realise that it is not a monitor's absolute size that is usually of interest, but rather its screen **resolution**. While a Web browser can manage to layout a document at any resolution, different resolutions do effect the layout and presentation of an HTML document. Resolution is measured in picture elements, called **pixels**. Typical monitor resolutions are 640x480, 800x600, 1024x870, 1280x1024 and 1600x1200.

Resolution and monitor size are independent of one another: a large monitor can have a low resolution, while a small monitor may have a high resolution. Resolution is determined by the hardware, the user, and the video card driver installed on the computer. A single monitor may have a choice of resolutions.

There is also the issue of a browser's 'live space'. Live space refers to the browser area the Web page is displayed in. This can vary from user to user, as the toolbars and status bars the user chooses to have displayed in the browser will reduce or increase the live space available to a Web page.

It is for all these reasons that the dichotomy between **fixed** and **flexible** Web page design has occurred.

By default, all Web pages are designed with flexibility in mind. Flexibility can be defined as a Web page's ability to resize and adapt to the available resolution, monitor and window sizes. Such an approach has both advantages and disadvantages.

Advantages:

- **Default Setting:** therefore no new tags are needed — the Web page fills entire space.
- **Philosophical:** flexibility is the philosophy of the Web i.e. it should be accessible by the greatest number of users.
- **Realistic:** resolutions, monitor and window sizes are always different. Keeping a Web page flexible allows it to be viewed on many available formats.

Disadvantages:

- **Uncomfortable:** reading text on large monitors is uncomfortable as the lines are too long.
- **Unpredictability:** the designer often cannot predict how a Web page will appear under varying resolutions and live space sizes.
- **Coherence:** on small monitors, everything may not appear correctly.

4.2.2 Flexible Design

While HTML is flexible by default, it should not be confused with thinking a flexible document is disorganised, poorly managed with an unstructured layout. A flexible HTML document can still be structured and organised by using, for instance, tables to create columns of text (as in newspapers), and provide layout design.

Flexible layout can be achieved by using percentage measurements for table dimensions. As an example, view the table below by changing the size of your browser's window (i.e. the live space). Observe that as the window size changes, so does the table size.

The table measurements used in this example are called 'relative' measurements, as the sizes are expressed in terms of a percentage of the screen space.

Books about Computing	
IBM PC Assembly Language and Programming	Abel, P
Object-Oriented Analysis and Design	Booch, G
Demonstration of a Flexible Table.	

4.2.3 Fixed Design

Fixed design expresses all dimensions in pixels: the dimensions remain fixed regardless of the size of the device it is viewed on. Such an approach has advantages and disadvantages:

Advantages:

- **Consistency:** it is usually important for companies to maintain a consistent image.
- **Control:** fixed design imposes restrictions on line length and hence stops uncomfortably long lines from occurring on Web pages.

Disadvantages:

- **Incompatible:** the chosen fixed size may be too large for a user's available live space, causing the user to scroll in order to view the whole page; a fixed Web page may also appear too small, leaving unsightly blank spaces.
- **Totalitarian:** the Web does not want to run the risk of being under too much control, i.e. we do not want every Web page to be identical. Some issues in print design are not transferable to the Web, which has its own idiosyncrasies, giving it the advantage of being independent of the print media.

To develop a fixed Web page using tables, supply all measurements in pixels. The table below is a demonstration

of a fixed table — change the size of your browser window and see the effect it has.

Books about Computing		
Title	Publisher	Author
IBM PC Assembly Language and Programming	Prentice-Hall	Abel, P
Object-Oriented Analysis and Design	Addison-Wesley	Booch, G
Demonstration of a Fixed Table		

To Do

Read up about standard table templates in your textbooks.

4.2.4 Activity 3: Using rowspan

This Activity introduces you to the attribute **rowspan**. The objective of this Activity is to create the following table.

red cell	silver cell
	gold cell

1. Let us start by creating the necessary code for displaying the silver and gold cells.

```
<TABLE style= "width: 30%" align = "center">
<tr >
    <td bgcolor = "silver" height =100> silver cell</td>
</tr>

<tr >
    <td bgcolor = "gold" height =100> gold cell</td>
</tr>
</TABLE>
```

2. Save this file as `adv_tab1.html` and view it in your browser. It should appear as so:

silver cell
gold cell

3. Now we insert a red cell spanning two rows. This is done with the **rowspan** attribute. The following syntax is used when designing tables that include **rowspan**.

`<td rowspan=x>` where **x** is the number of rows to be spanned.

Reopen `adv_tab1.html` and make the amendment shown in bold, below.

```
<TABLE style= "width: 30%" align = "center">
<tr >
  <td bgcolor = "red" rowspan = 2 width = 75> red cell</td>
  <td bgcolor = "silver" height =100> silver cell</td>
</tr>
<tr >
  <td bgcolor = "gold" height =100> gold cell</td>
</tr>
</TABLE>
```

4. Save this exercise as `adv_tab2.html` and view it in your browser. It should now look as below:

red cell	silver cell
	gold cell

4.2.5 Activity 4: Using colspan, cellspacing and cellpadding

This Activity introduces you to the attributes **colspan**, **cellspacing** and **cellpadding**. The objective of this Activity is to create the following table.

red cell	
silver cell	gold cell

1. Let us begin by creating the silver and gold cells.

```
<TABLE style= "width: 30%" align = "center">
<tr >
    <td bgcolor = "silver" height =100> silver cell</td>
    <td bgcolor = "gold" height =100> gold cell</td>
</tr>
</TABLE>
```

2. Save this file as adv_tab3.html and view it in your browser. It should appear as below:



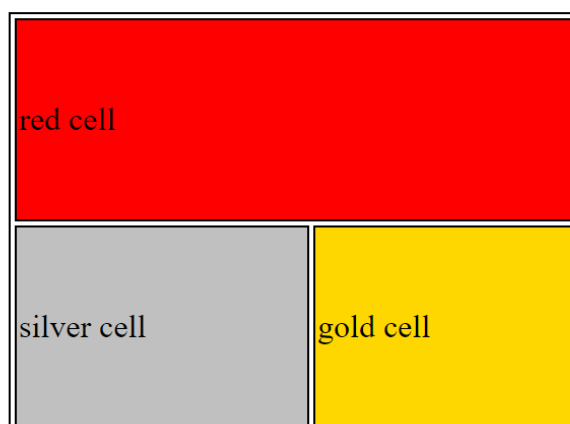
3. For the next step we insert a cell that spans the two columns, using the **colspan** attribute:

`<td colspan=x>` where **x** is the number of columns to be spanned.

Reopen adv_tab3.html and make the amendment as shown in bold, below:

```
<TABLE style= "width: 30%" align = "center">
<tr>
<td colspan=2 height=100 bgcolor="red">red cell</td>
</tr>
<tr >
    <td bgcolor = "silver" height =100> silver cell</td>
    <td bgcolor = "gold" height =100> gold cell</td>
</tr>
</TABLE>
```

4. Save this exercise as adv_tab4.html and view it in your browser. It should appear as below.



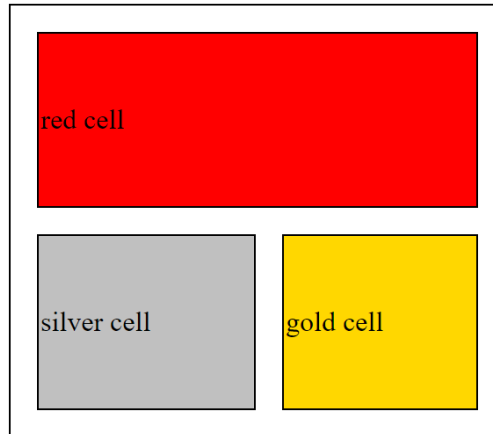
5. The space between the cells is known as the cellspacing. This is controlled with the table attribute **cellspacing**. The following syntax is used with **cellspacing**:

`<table cellspacing=x>` where **x** is the amount of cellspacing required.

Reopen adv_tab4.html and make the following alterations to the code, as shown in bold.

```
<TABLE style= "width: 30%" align = "center" cellspacing = 15>
<tr>
<td colspan=2 height=100 bgcolor="red">red cell</td>
</tr>
<tr >
<td bgcolor = "silver" height =100> silver cell</td>
<td bgcolor = "gold" height =100> gold cell</td>
</tr>
</TABLE>
```

6. Save this exercise as adv_tab5.html and view it in your browser. It should appear as below:



7. The space between the text 'red cell' and the cell's border is known as the cellpadding. This can be altered with the attribute **cellpadding**:

`<table cellpadding = x>` where **x** is the thickness, measured in pixels, of the desired cellpadding.

Reopen adv_tab5.html and make the following amendments to the code, shown in bold.

```
<TABLE style= "width: 30%" align = "center" cellspacing = 2>
<tr>
<td colspan=2 height=100 bgcolor="red">red cell</td>
</tr>
<tr >
<td bgcolor = "silver" height =100> silver cell</td>
<td bgcolor = "gold" height =100> gold cell</td>
</tr>
</TABLE>
```

8. Save this exercise as adv_tab6.html and view it in your browser. It should appear as required.

red cell	
silver cell	gold cell

4.2.6 Activity 5: More cellspacing and cellpadding

The objective of this Activity is to create the table shown below.

This Activity also introduces some of the anomalies that Web page developers deal with during Web page design. In particular, we focus on anomalies with the cellspacing attribute.

Three points should be noted for the above table:

- There are no visible borders, therefore **border=0** is needed.
- There is no cellspacing between cells, so **cellspacing=0**.
- There is no cellpadding, therefore **cellpadding=0**. Now follow these steps to complete the activity.

1. Begin a new file in a text editor and enter the following HTML code:

```
<TABLE style= "width: 30%" height = 30 align = "center" cellspacing = 0
cellpadding = 0>
  <tr>
    <td height=15 bgcolor="darkred"></td>
    <td bgcolor="red"></td>
    <td bgcolor="pink"></td>
  </tr>
  <td height=15 bgcolor="darkblue"></td>
  <td bgcolor="blue"></td>
  <td bgcolor="lightblue"></td>
</TABLE>
```

2. Save your file as adv_tab7.html and view it in your browser. It should appear as required.

To Do

Read up about cell spacing in your textbooks. Try to discover common problems and mistakes that Web designers encounter when using tables. While doing this, continue to build on your list of table related tags.

4.2.7 Activity 6: Fixed and flexible Web page design

The objective of this Activity is to compare fixed and flexible Web page design.

1. Design a table with two columns, each containing text.
2. First create the table using a flexible page layout. Ensure that your table works as expected by resizing the browser window.
3. Now create the table using a fixed Web page design. Again, check that your answer is correct by resizing the browser window.

4.2.8 Activity 7: Time Table

Write the necessary HTML code for your own study timetable. This should look similar to the one shown below. (Hint: use the **colspan** and **rowspan** attributes).

Ian's Timetable

		Morning			lunch	afternoon			evening			
		9-10 am	10-11 am	11-12 am	1-2 pm	2-3 pm	3-4 pm	4-5 pm	5-6 pm	6-7 pm	7-8 pm	8-9 pm
Work	Monday	Development Meeting				Client Meeting			commute	Free		
	Tuesday	in Office				in Office						
Lecture	Wednesday	CSC205				preparation						
	Thursday	MAM200				Tutorials for MAM200			Free			
Research	Friday	Research				Research						

4.3 Review Questions

Use the following questions to assess your understanding of HTML tables. Compare your answers to those given below.

1. Initially, why were tables introduced?
2. Why do Web designers prefer to use tables instead of frames?
3. Make a list of possible instructions to develop a table in HTML.
4. Define the difference between fixed and flexible Web page design. State their advantages and disadvantages.
5. You can find the answers at the end of the chapter.

4.4 Discussions and Answers

4.4.1 Correct code for Activity 1 step 1

The code should appear as below, with the `</TITLE>` tag before the `</HEAD>` tag

```
<HTML>
<HEAD>
<TITLE>
  HTML Table Design
</TITLE>
</HEAD>
```

```
<BODY>
</BODY >
</HTML >
```

4.4.2 Solution to Activity 1 step 5

```
<HTML>
<HEAD>
<TITLE>
  HTML Table Design
</TITLE>

<style>
table, th, td {
  border: 1px solid black;
}
</style>

</HEAD>
<BODY>

  <TABLE style= "width: 80%">
  <tr>
    <TD width=25% height=70 bgcolor="red"> red cell </td>
    <TD width=75% bgcolor="lightblue"> light blue cell </td>
  </tr>
  </TABLE>

</BODY >
</HTML >
```

4.4.3 Solution to Activity 2: HTML Color Table

Here is the code used to create the table in Activity 2.

```

<HTML>
<HEAD>
<TITLE>
    HTML Table Design
</TITLE>

<style>
table, th, td {
    border: 1px solid black;
}
</style>

</HEAD>
<BODY>

    <TABLE style= "width: 80%" align = "center">
    <caption> Some HTML colors </caption>
    <tr >
        <th width = 150> Color</th>
        <th width = 150> Name</th>
        <th width = 150> Hexadecimal</th>
        <th width = 150> RGB Value</th>
    </tr>
    <tr >
        <td bgcolor = "fa8072"> </td>
        <td > Salmon</td>
        <td > fa8072 </td>
        <td > 250-128-114 </td>
    </tr>
    <tr >
        <td bgcolor = "ffd700"> </td>
        <td > Gold</td>
        <td > ffd700 </td>
        <td > 255-215-0 </td>
    </tr>
    </TABLE>

</BODY >
</HTML >

```

4.4.4 Discussion Topic

The purpose of this exercise is to give you the opportunity to discuss the subject of fixed and flexible page design with other students in the course.

Before going to the Discussion Forum, do the following.

1. Go to the O'Reilly's Web site [<http://www.oreilly.com>] and the UCT website [<http://www.uct.ac.za>] and
 - Decide if these pages use fixed or flexible;
 - Consider whether there are any disadvantages with this type of Web page design.
2. Find another example of a Web page that uses a fixed page design, and another that uses a flexible page design.

You are now ready to join the Discussion Forum. In the forum you should be able to:

- Discuss your thoughts on the design principles used to create the O'Reilly and UCT websites
- Share the Web page design examples that you have found and discuss the advantages and disadvantages of fixed and flexible Web page design.

Additional Resources

You may find these online resources useful in your study of tables.

- Table Tutorial [<http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimerPrintable.html#TA>]
- HTML 4.0 Specification Tables Section [<http://www.w3.org/TR/REC-html40/struct/tables.html>]

4.4.5 Answers to Review Questions

1. Tables were initially developed as a tool to organise and display data in columns and rows, but they are now also use to support Web page design.
2. Designers prefer to use tables to frames because older versions of browsers do not support frames.
3. List of possible instructions to develop a table in HTML:
 - a. Insert the table tag and decide on its dimensions.
 - b. Add a row with the TR tag.
 - c. Insert a cell in the newly created row, with dimensions and other characteristics.
 - d. Add the data to be displayed.
 - e. Close the data cell.
 - f. Repeat steps three through five as necessary.
 - g. Terminate the row.
 - h. Return to the second step and repeat until all of necessary rows have been added.
 - i. Terminate the table.
4. Fixed Web page design gives designers more control over the Web page, as they can define the exact dimensions of the layout of the page. Flexible Web page design uses relative measurements, and so the element sizes may change, depending on the window size. The advantages of a fixed page design: constant, consistent look to the page; controls line length. Disadvantages of a fixed page design: design may be too large and cause scrolling on the screen. Advantages of a flexible page design: Web page takes up whole page, meets the needs of all resolutions, monitor and window sizes. Disadvantages of a flexible page design: can create unreadable line lengths; possibly unpredictable design.