LECTURE 9:
PAGE LAYOUT REVISED

Liquid

Fixed

Elastic

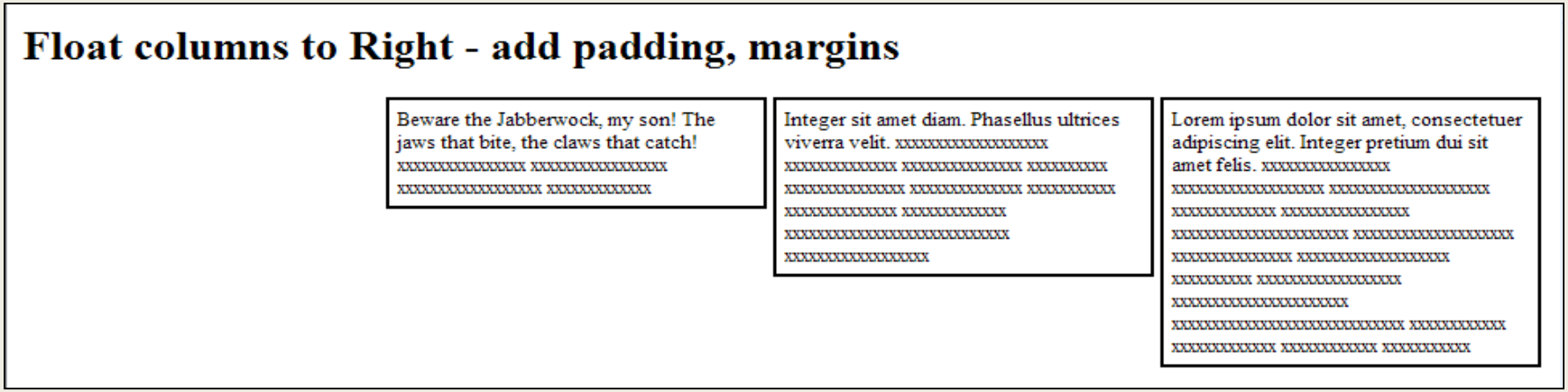
Content of Lecture

- Creating sidebars, links, multi-column page layouts (Liquid layout)
- Liquid/Fixed layout

LIQUID PAGE LAYOUT

Create a Multi-Column Floating Layout

- Create multiple divs, each with a float & width attribute.
- As the columns float, the paragraph following the columns is given a clear value to make sure it is placed below the columns

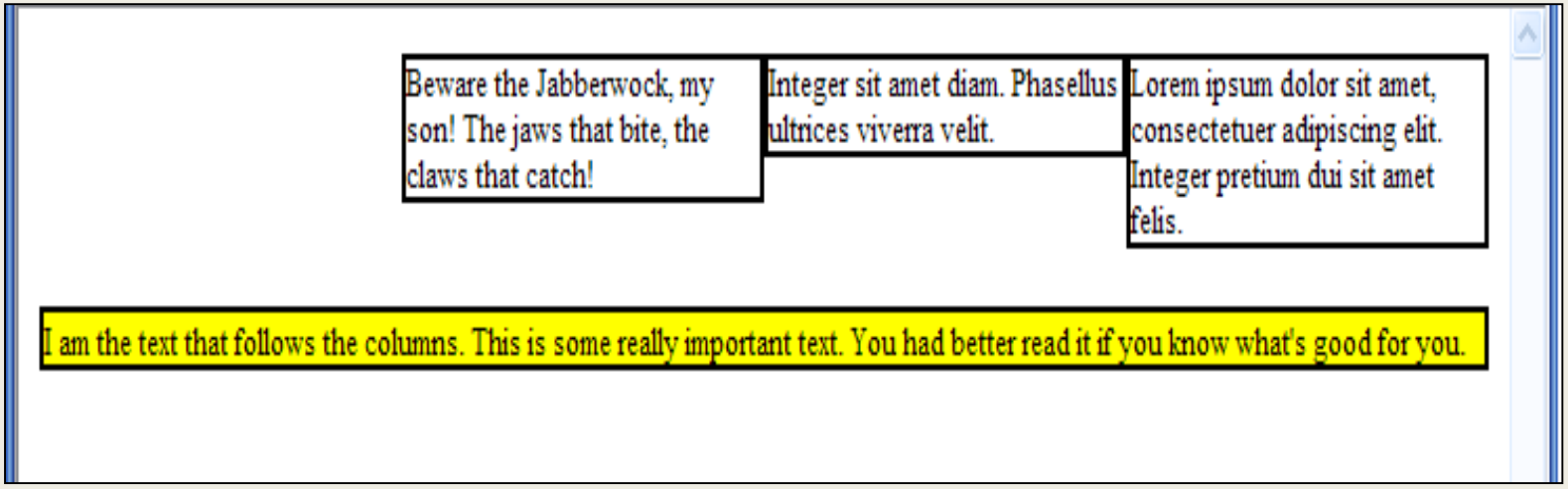


Multi-Column Floating Layout

```
<div class="column">
  Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Integer pretium dui sit amet felis.
</div>

<div class="column">
  Integer sit amet diam. Phasellus ultrices viverra velit.
</div>

<div class="column">
  Beware the Jabberwock, my son!The jaws that bite, the claws that catch!
</div>

<p id="aftercolumns">
  I am the text that follows the columns.This is some really important text. You had better read it if you know what's good for you.
</p>
```

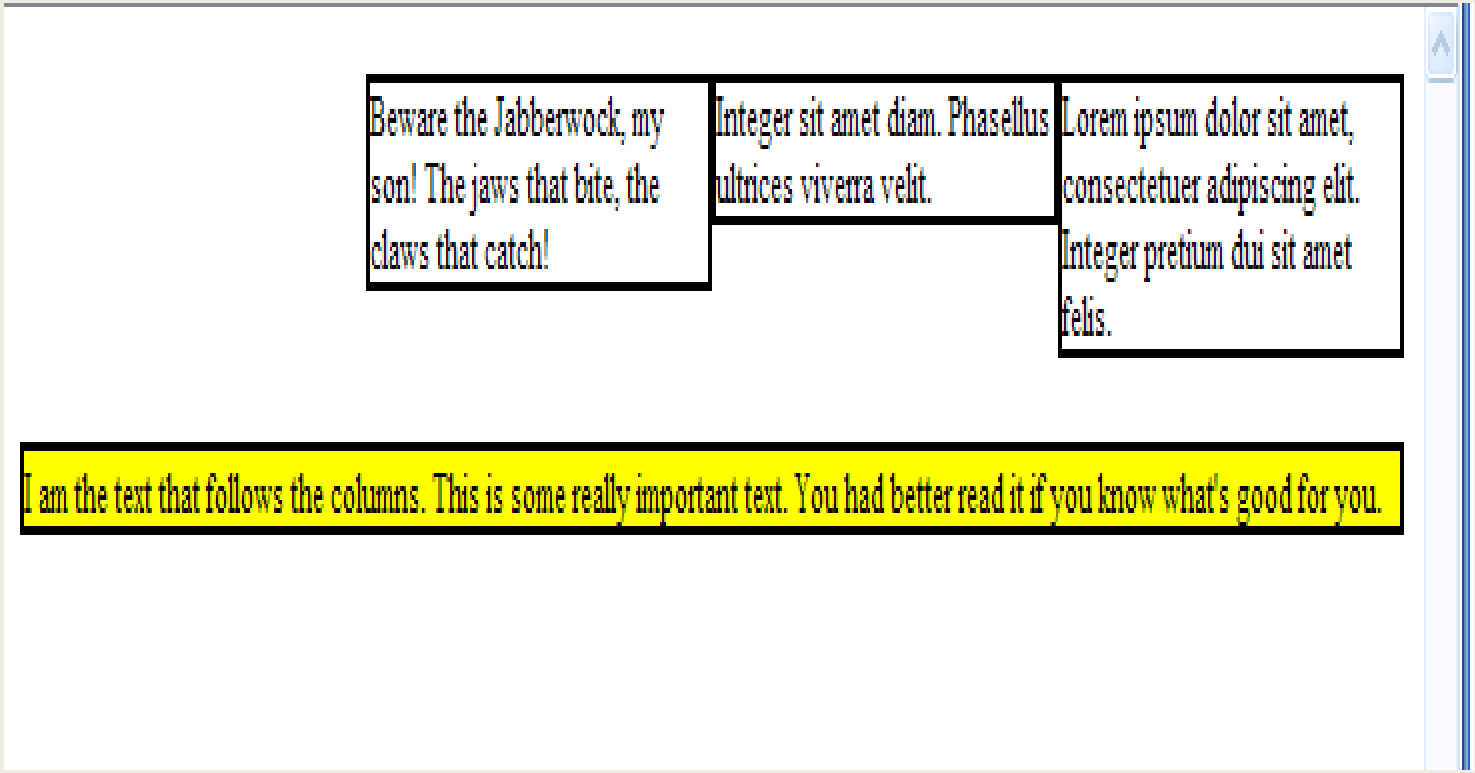
```
<style >
div, p {
  border: 2px solid black;
}
.column {
  float: right;
  width: 25%;
}
#aftercolumns {
  background-color: yellow;
  clear: both;
}
</style>
```

Multi-Column Floating Layout

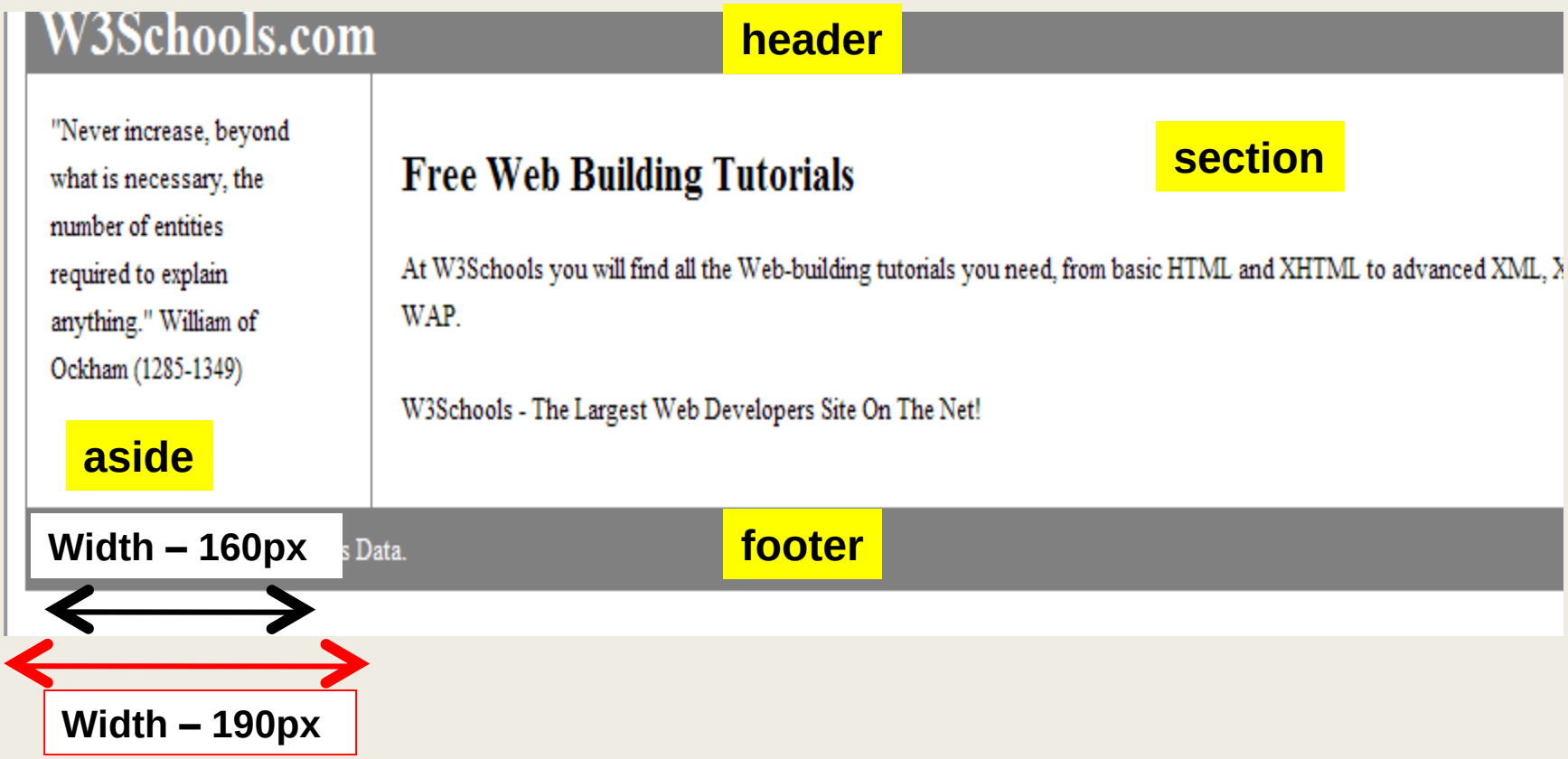


- Important that any content to follow the columns should clear the previously set float.
- If not done, the final paragraph in the example will wrap around the floating columns.

Multi-Column Floating Layout



Using Float – Page Layout



Recap-HTML5 layout tags

Using Float – Page Layout (Liquid Layout)

- Use HTML5 tags + divs
  - *Container (or wrapper) div*
  - *Header*
  - *Aside*
    - Float:left
  - *Section*
  - *Footer*

Using Float – Page Layout

```
<body>
<div class="container">
    <header>
        <h1>W3Schools.com</h1>
    </header>
        <aside><p>"Never increase, beyond what is necessary, the
        number of entities required to explain anything." William of
        Ockham (1285-1349)</p>
        </aside>
    <section>
        <h2>Free Web Building Tutorials</h2>
        <p>At W3Schools you will find all the Web-building tutorials you
        need, from basic HTML and XHTML to advanced XML, XSL,
        Multimedia and WAP.</p>
        <p>W3Schools - The Largest Web Developers Site On The
        Net!</p>
    </section>
    <footer>Copyright 1999-2005 by Refsnes Data.
    </footer>
</div>
</body>
```

Using Float – Page Layout

- Aside
  - *Float it to the left*
  - *Width of 160px*
  - *No margins, padding all round of 1rem*
- Section
  - *Left margin of 190px*
  - *Left border of 1px, solid line and grey*
  - *Padding all round of 1rem*
- Header & Footer
  - *Padding of 0.5rem*
  - *Text in white, grey background*
  - *Clear the left float*

Note:
<div class =
“container”>

This is wrapped
around
all the content

Using Float – Page Layout

```
<style>
.container
{
        width:100%;
        margin:0px;
        border:1px solid gray;
        line-height:150%;
}
header,footer
{
        padding:0.5rem;
        color:white;
        background-color:gray;
        clear:left;
}
```

```
header{
        padding:0;
        margin:0;
}
aside{
        float:left;
        width:160px;
        margin:0;
        padding:1rem;
}
section{
        margin-left:190px;
        border-left:1px solid gray;
        padding:1rem;
}
</style>
```

Using Float – Page Layout

- Note: previous example: defined the page width as a %
- If we use % for the width of a page, we are creating a LIQUID page layout.
- If you reduce the size of the page, the content wraps around the available space and acquires the shape of the available space.

HORIZONTAL MENUS

Use Float – Horizontal Menu

- Unordered list of items containing lists created
- List contained within <nav> </nav> tags
- All unwanted spacing and bullets are removed
- Style <li> elements as inline elements. Remove the bullet.
- Style <a> links as blocks with padding etc.
- Set the <a> link to float to the left.
- Each hyperlink in the list has a width of 6rem (6 times the size of the current font).
- Add some colors and borders
- Style for **hover**, **active**, **current page** etc.

16

Use Float – Horizontal Menu

```
<nav>
    <ul >
        <li><a href="#">Link one</a></li>
        <li><a href="#">Link two</a></li>
        <li><a href="#">Link three</a></li>
        <li><a href="#">Link four</a></li>
    </ul>
</nav>
```

```
nav ul {
padding:0;
margin:0;
}

nav li {
display:inline;
list-style-type:none;
}
```

```
nav a {
    display:block ;
    float:left;
    width:6rem;
    text-decoration:none;
    color:white;
    background-color:purple;
    padding:0.2em 0.6em;
    border-right:1px solid white;
}
a:hover {
    background-color:orange;
}
```


17

Styling Current Page

- Assign an id to page that is the current page

```
<li><a href = " # " id ="current_page">Home</a></li>
```

- Do this for each page, in turn
- In CSS, style the current page so that the link in the navigation bar is a different colour to the other links

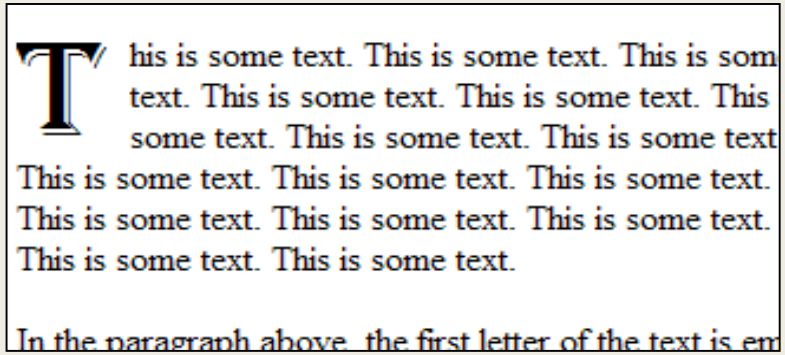
```
#current_page{
    background-colour, font-colour, font-style etc.
}
```

FLOATING A LETTER

Floating a Letter

- Example:
- First letter of a paragraph float to the left and style the letter
- Embed the first element of the text in a element

```
span
{
float:left;
width:0.7em;
font-size:400%;
font-family:algerian,courier;
line-height:80%;
}
```



20

Page Layout

Problem:

- No way of knowing how wide, skinny, tall or short a user's browser window will be.
  - *User's set browser window to fill the monitor at a standard screen resolution or*
  - *Set to some comfortable resolution*
  - *Precise size of a given web page is unknown*
- Text size - how large or small will it be?

Page Layout

Solution

- Three general layout approaches have emerged to deal with these problems:
  1. *Liquid Page Layout*
  2. *Fixed Width Page Layout*
  3. *Elastic Layout*

1. Liquid Page Layout

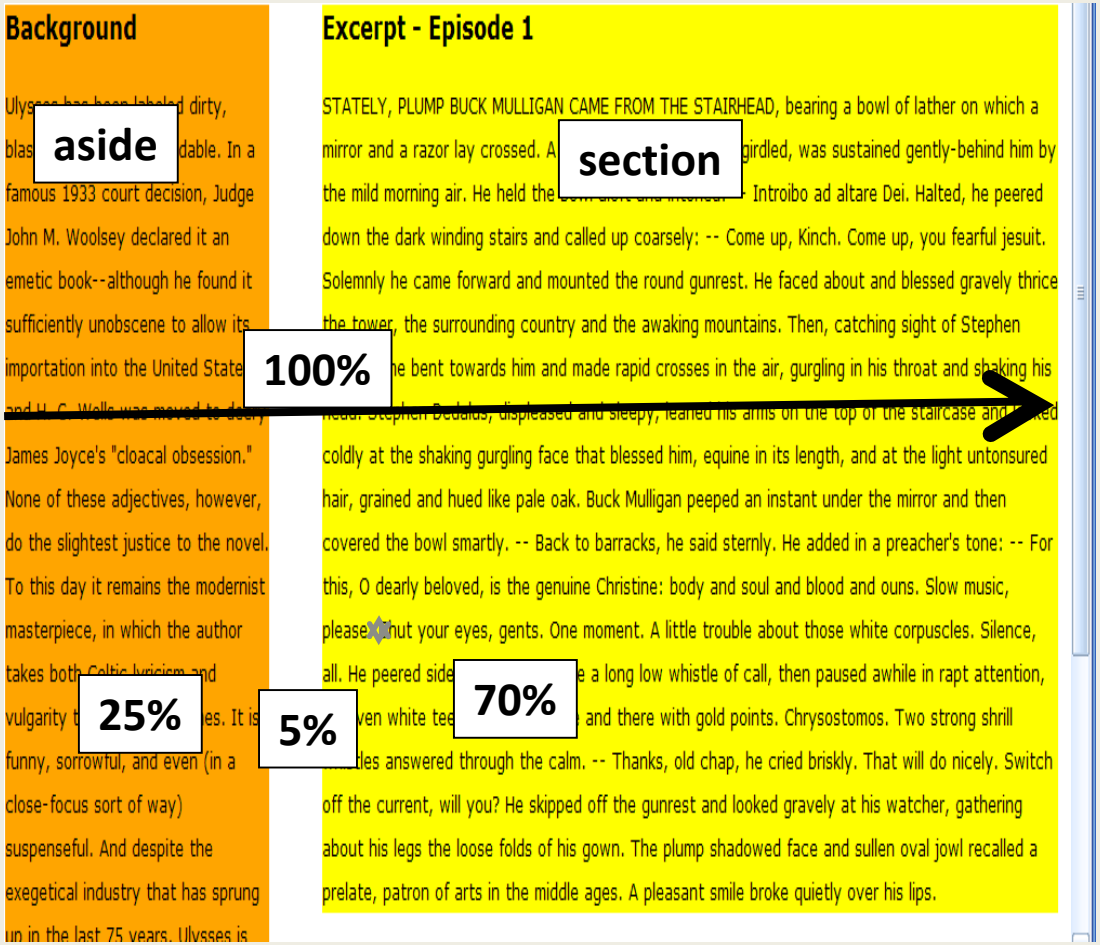
- Also known as *Fluid Layout*
- Created by specifying width in percentage % values
- Also, if you do not specify the width at all – width will be set to auto, by default
  - *Element will fill the available width of the window or other containing element*

1. Liquid /Fluid Layout

```
body{
        font-family: Verdana, Helvetica,
                                sans- serif;
        font-size: small;
        line-height: 175%;
}
aside{
        width: 25%;
        float: left;
        background-color: orange;
}
section{
        width: 70%;
        margin-left: 5%;
        float: left;
        background-color: yellow;
}
```

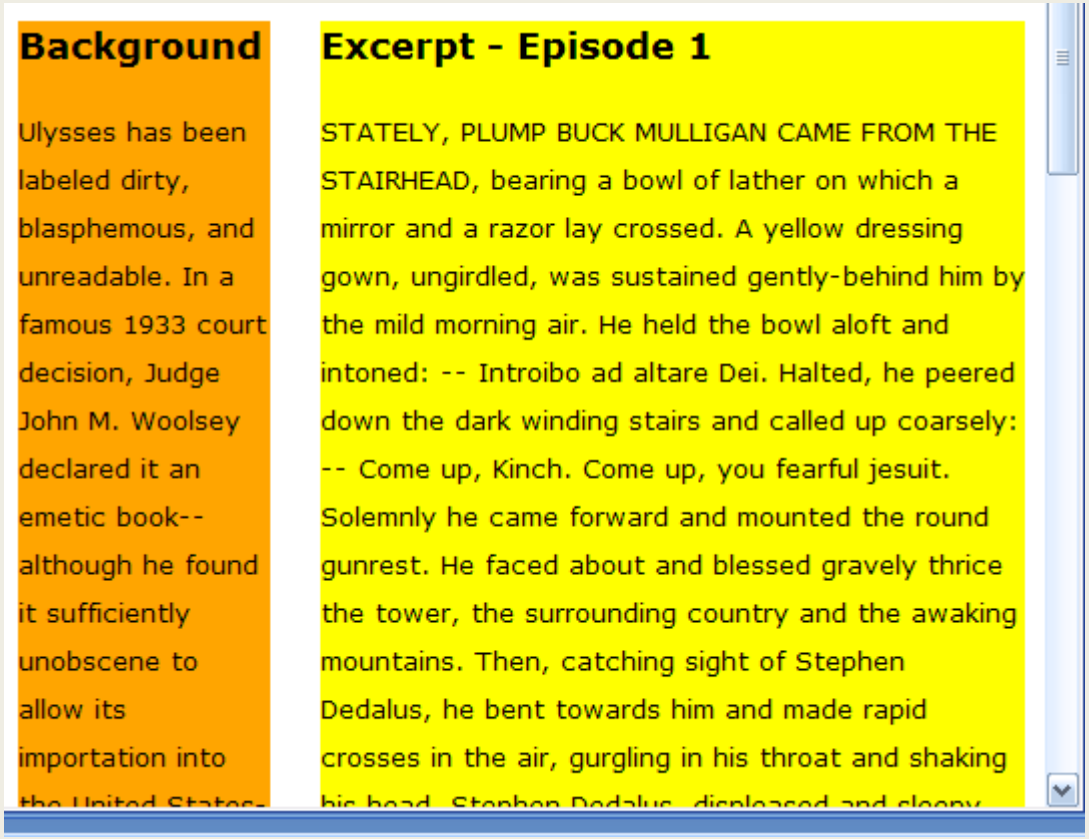
```
<body>
<h1>Lecture 3: Layouts</h1>

<aside>
        <h3>Background</h3>
        ……….
</aside>

<section>
        <h3>Excerpt - Episode
1</h3>
        ……..
</section>
</body>
```

1. Liquid /Fluid Layout



Content resizes when the browser is resized

1. Liquid /Fluid Layout



Content resizes when the browser is resized

1. Liquid /Fluid Layout

- NOTE: Page needs margins and padding to be added in order to look well, as the text is too close to the edge of the coloured section.

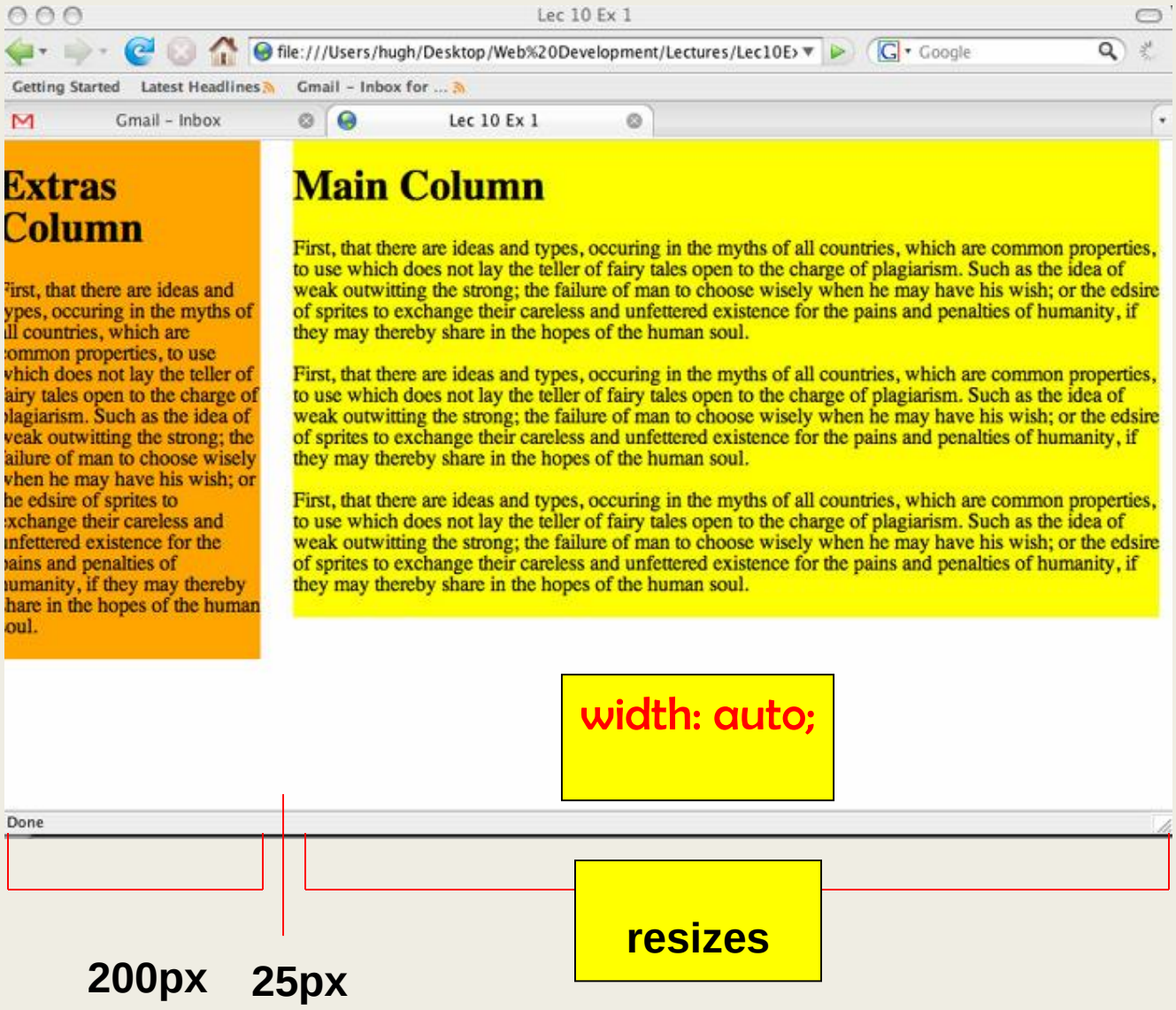
Liquid Layout – Variation 2

- Variation of Liquid Layout example
- Set Left-hand column to a fixed width and then set the width of the main content column to auto so it fills the remaining space

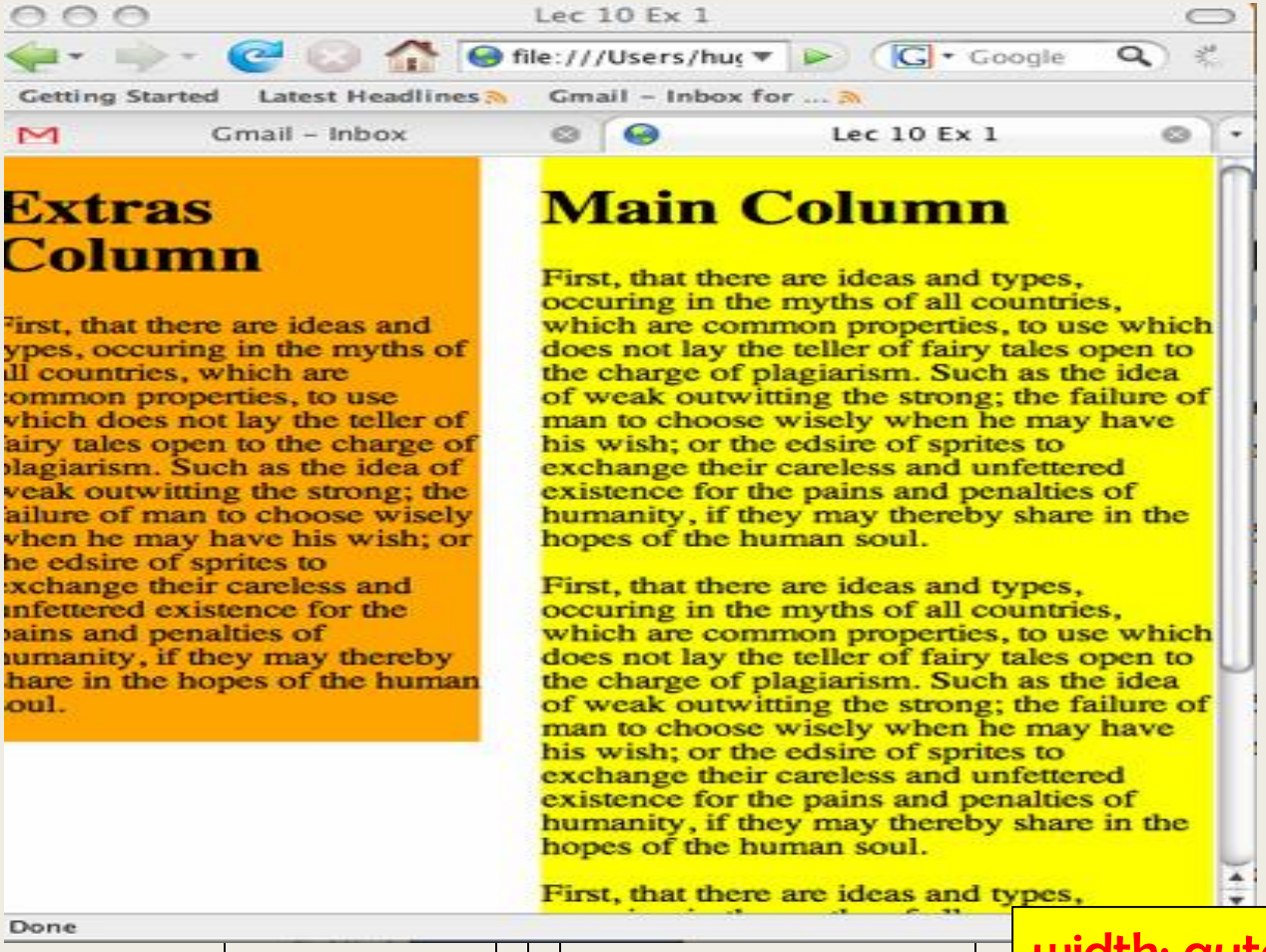
Liquid Page Design

```
section {
    width: auto;
    position: absolute;
    top: 0;
    left: 225px;
    background-color: yellow;
}
aside {
    width: 200px;
    position: absolute;
    top:0;
    left:0;
    background-color: orange;
}
```

Liquid Page Design



Liquid Page Design



2. Fixed Layout

- Designer sets a page to a specific pixel width
- Based on traditional guiding principles of graphic design e.g. constant grid, relationship of page elements and comfortable line lengths.
- Need to select a page width based on common monitor resolutions
  - *Most designed for 800 X 600 pixel browser*
  - *1000 pixel width becoming more common*

2. Fixed Layout

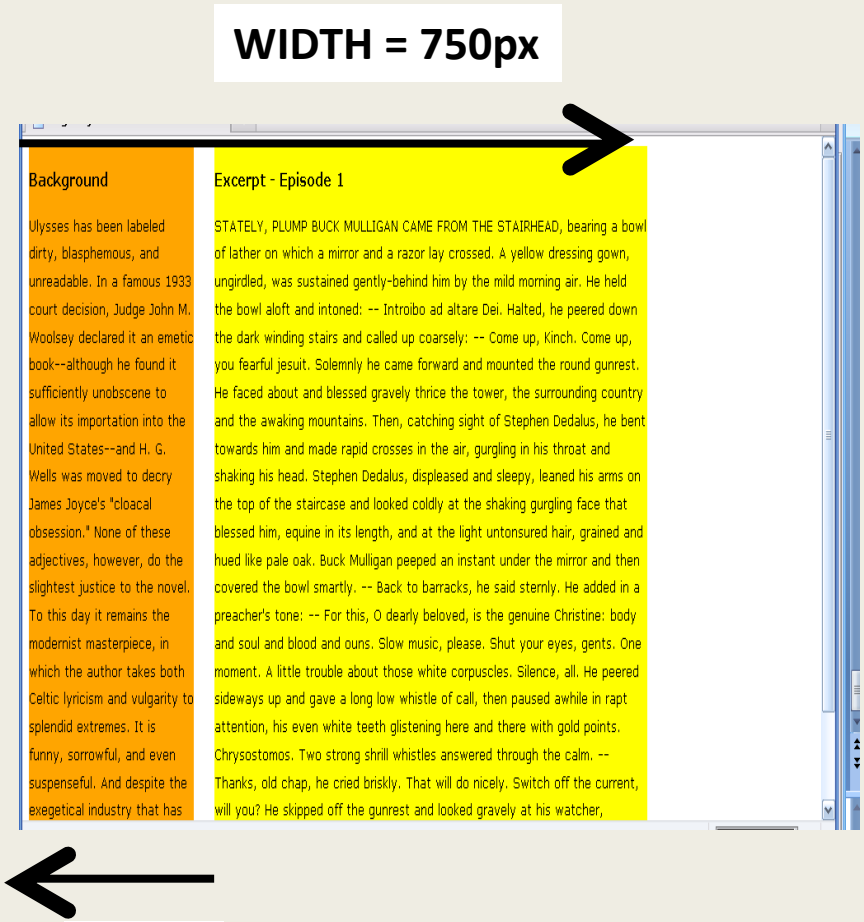
- Need to decide where to position the fixed-width layout in the browser window
- Default position: left edge of browser window (viewport) , with extra space to the right
- Some designers centre the page, thereby splitting the extra space to the right and left.

2. How to create Fixed Layout

- Content of the entire page is put into a DIV
- DIV is then set to a specific pixel width
- DIV often called “wrapper”, “container”, “content”, or “page”

```
<body>
 <div id="wrapper">

      <aside>
       </aside>

      <section>
      </section>

 </div>
</body>
```

2. How to create Fixed Layout



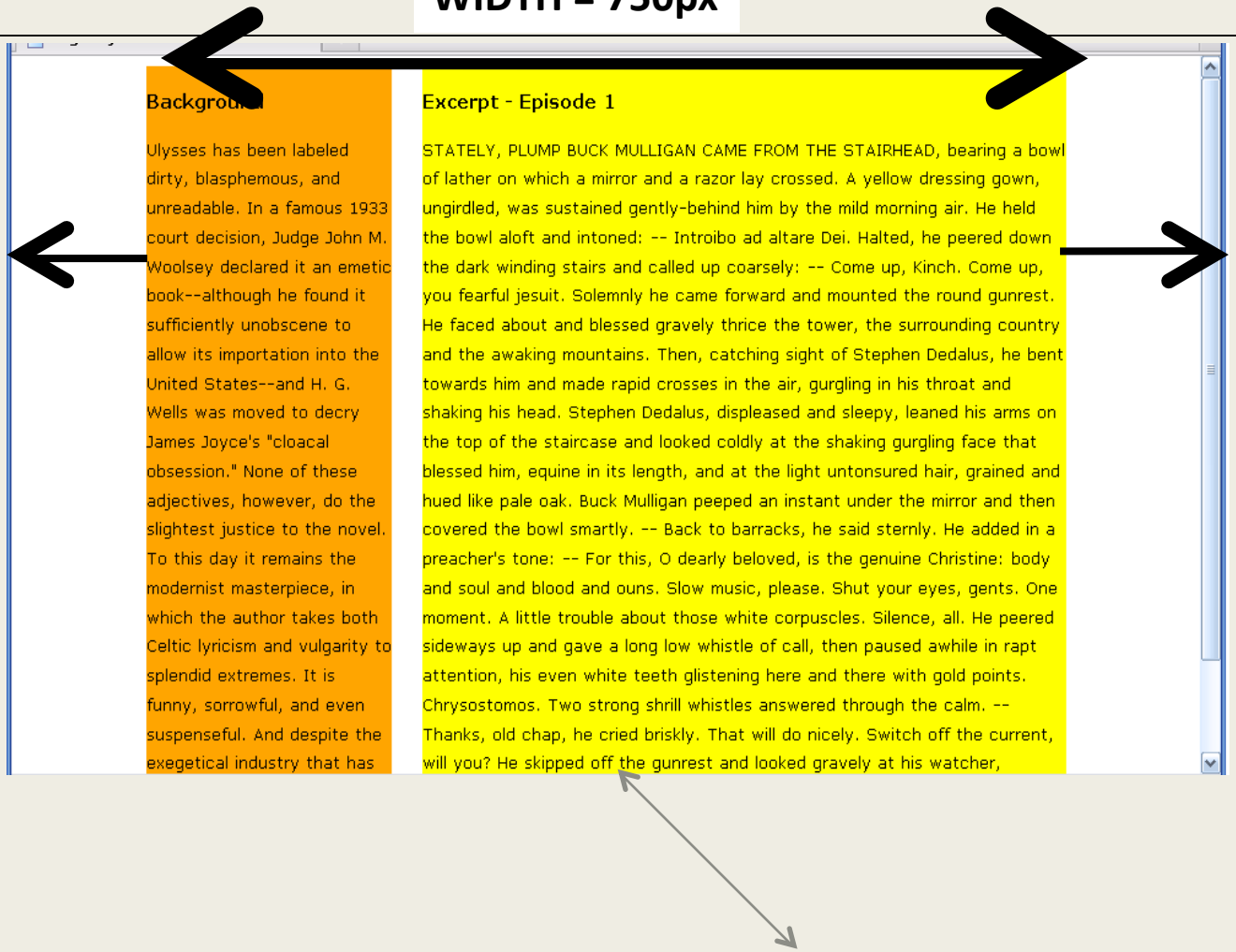

```
#wrapper{
        width: 750px;
        position: absolute;
        padding: 0px;
}
aside{
        position: absolute;
        top: 0px;
        left: 0px;
        width: 200px;
        background: orange;
}
section{
        position: absolute;
        margin-left: 225px;
        background: yellow;
}
```

2. How to create Fixed Layout – Centred

- DIV may also be centered in the browser window by setting the
  - *margin-right and margin-left properties to auto.*
- Position of this outer DIV must be set to
  - ***RELATIVE**, in order to centre the contents.*

2. Fixed Layout – Centred

WIDTH = 750px



2. Fixed Layout – Centred

```
<body>
 <div id="wrapper">

      <aside>
       </div>

      <section>
      </div>

 </div>
</body>
```

```
#wrapper{
        width: 750px;
        position: relative;
        padding: 0px;
        margin-right: auto;
        margin-left: auto;
}

aside{
        position: absolute;
        top: 0px;
        left: 0px;
        width: 200px;
        background: orange;
}

section{
        position: absolute;
        margin-left: 225px;
        background: yellow;
}
```

3. Elastic Layout

- Elastic layouts expand or contract with the size of the text
- Result: Line length (in terms of words or characters per line) stay the same regardless of text size – predictable line length
- Becoming more popular as an approach

3. Elastic Layout

- Advantage over Liquid Layout
  - *where line lengths can become too long*
- Advantage over Fixed Layout
  - *where large text may cause awkward looking layouts with few words / characters per line*

3. How to create Elastic Layout

- Key – Use em units of measurement for all dimensions
- Use ems for the dimensions of containing elements.
- Em unit is based on the size of the text
- E.g.Element with 12-pixel text; an em is 12 pixels
- Better to set font sizes in ems because it allows text to be resized with the zoom feature on browsers.

3. Elastic Layout – Ems

- Text size set to 76% = 12 pixels on most browsers
- Page set to 40em
- Resulting page = 40em X 12px/em = 480 px
- If the user resizes the text to 24 pixels, the page grows to 960 pixels
  - *Getting close to available canvas space on browsers with 1024 pixel wide monitors*

3. How to create Elastic Layouts

```
body{
        font-family: Verdana, Helvetica,
                                sans-serif;
        font-size: 80%;
        line-height: 150%;
}

#wrapper{
        width: 60em;
        position: relative;
        padding: 0px;
        margin-right: auto;
        margin-left: auto;
}
```

```
aside{
        position: absolute;
        top: 0px;
        left: 0px;
        width: 15em;
        background: orange;
}

section{
        position: absolute;
        margin-left: 18em;
        background: yellow;
}
```

Elastic Layout

CSS3 – Create Multiple Columns

How many columns?

Note: Internet Explorer 9, and earlier versions, does not support the column-count property.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

```
<body>
<p><b>Note:</b> Internet Explorer 9,
    and earlier versions, does not
    support the column-count
    property.</p>
<div class="newspaper">
Lorem ipsum dolor sit
    .....
</div>
</body>
</html>
```

```
.newspaper
{
-webkit-column-count:3; /*
    Safari and Chrome */
column-count:3;
}
```

The **column-count** property specifies the number of columns a page should be divided into.

CSS3 – Multiple columns

How many columns????

Note: Internet Explorer 9, and earlier versions, does not support the column-rule property.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

3 columns
Gap
Line

CSS3 – Multiple columns

```
.newspaper
{
-webkit-column-count:3; /* Safari and Chrome */
column-count: 3;
-webkit-column-gap:40px; /* Safari and Chrome */
column-gap: 40px;
-webkit-column-rule:4px outset purple; /* Safari and Chrome */
column-rule: 4px outset purple;
}
```

The **column-gap** property specifies the gap between the columns

The **column-rule** property sets the **width, style, and color** of the rule between columns – similar to **border** property

CSS3 – Multiple Column Properties

column_count	Number of columns an element should be divided into
column-gap	Size of gap between the columns
column-rule	Specify the width, style and colour of a line between the columns (like border property)
column-span	No of columns an elements e.g. <h1> can span – 1 or ALL
column-width	Specify the width of the columns
columns	Shorthand for specifying column-width & column-count

Page Layout

- Many approaches to web page layout
- Approach you choose will be dictated by:
  - *Purpose of web page*
  - *Content*
  - *Intended Audience*
  - *Etc.*
- **Web developers need a toolbox of page layout strategies to apply quickly to see how layout looks**
- Will look at a number of page layout templates
  - *2 /3 columns with footers using floats*
  - *2/3 columns using positioning (liquid)*
  - *3 column with header and footer using positioning (fixed)*

Page Layout Templates

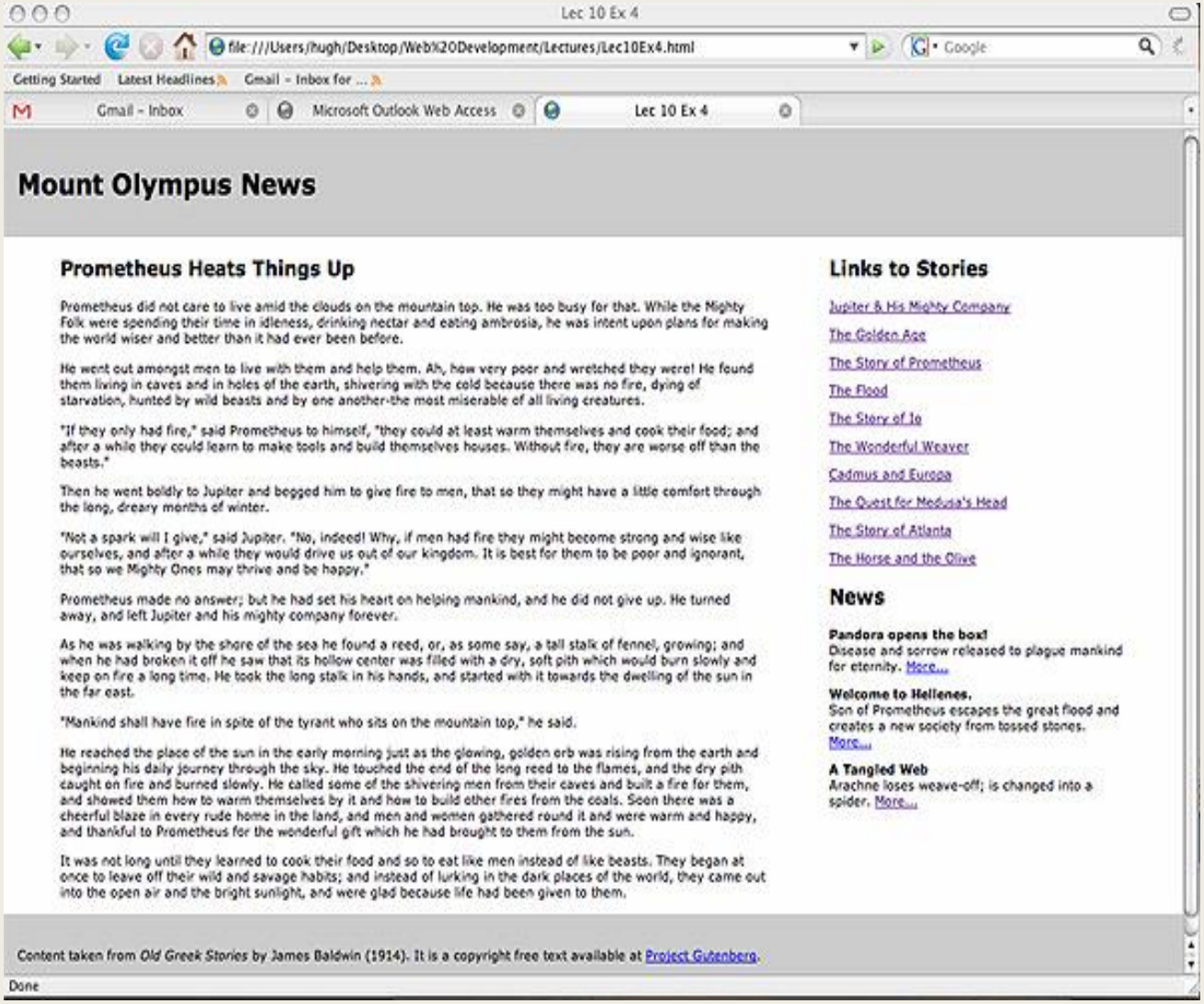
- The following slides present a series of page layout templates.
- They comprise of a variety of two and three column layouts that you can use as a basis for your pages.
- Each of these layouts either uses *floats* or *positioning*.
- We will see that both *fixed* and *liquid* layouts are possible in each case.
- Web developer should have templates of various web page layouts at their disposal, so that they can use them, as required

Multicolumn Layouts Using Floats

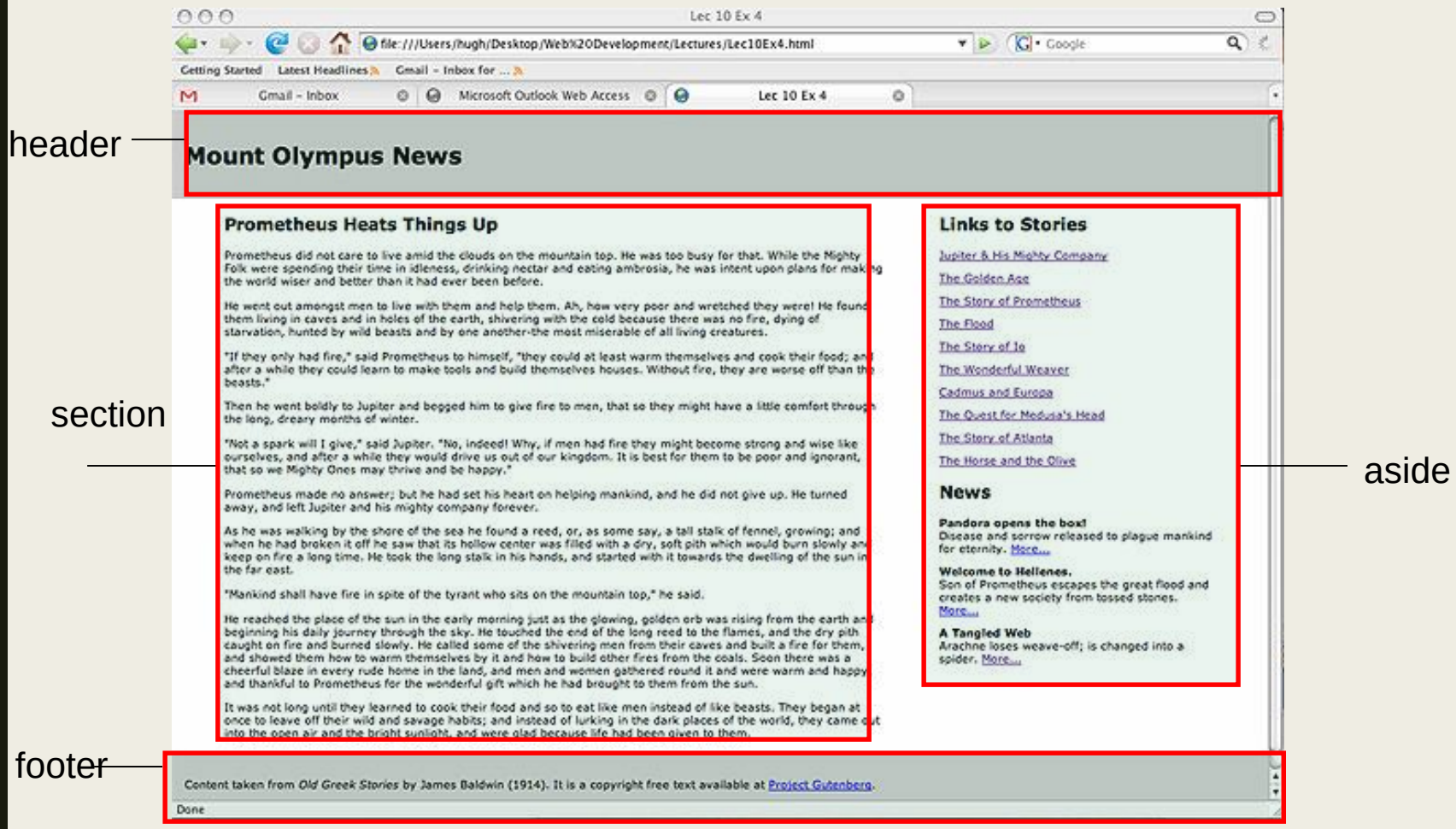
- **Template 1**
- Two Column with Footer
- Method: FLOAT
- Layout: LIQUID

- This template creates a classic two column layout with a header across the top, containing the page title.
- The left hand column contains the main text and then the right hand column contains a list of links and some news items.
- A footer is also provided at the bottom.

Multicolumn Layouts Using Floats



Multicolumn Layouts Using Floats



Multicolumn Layouts Using Floats

```
header {
        background-color: #CCC;
        padding: 15px;}
section {
        float: left;
        width: 60%;
        margin: 0 5%;
        outline: 2px dashed #0053ae; }
aside {
        float: left;
        width: 25%;
        margin: 0 5% 0 0;
        outline: 2px dashed #dd0009;}
footer {
        clear: left; /* starts the footer below the floated content */
        padding: 15px;
        background: #CCC;}
```

Float section to the left

Make it 60% of the browser width

Multicolumn Layouts Using Floats

- **Template 2**
- Three Column with Footer
- Method: FLOAT
- Layout: FIXED

- This template creates a classic three column layout with a header across the top, containing the page title.
- The left hand column contains a list of links, the middle section contains the main content and then the right hand column some news items.
- A footer is also provided at the bottom.

Multicolumn Layouts Using Floats

Mount Olympus News

Links to Stories

Jupiter & His Mighty Company

The Golden Age

The Story of Prometheus

The Flood

The Story of Io

The Wonderful Weaver

Cadmus and Europa

The Quest for Medusa's Head

The Story of Atlanta

The Horse and the Olive

Prometheus Heats Things Up

Prometheus did not care to live amid the clouds on the mountain top. He was too busy for that. While the Mighty Folk were spending their time in idleness, drinking nectar and eating ambrosia, he was intent upon plans for making the world wiser and better than it had ever been before.

He went out amongst men to live with them and help them. Ah, how very poor and wretched they were! He found them living in caves and in holes of the earth, shivering with the cold because there was no fire, dying of starvation, hunted by wild beasts and by one another-the most miserable of all living creatures.

"If they only had fire," said Prometheus to himself, "they could at least warm themselves and cook their food; and after a while they could learn to make tools and build themselves houses. Without fire, they are worse off than the beasts."

Then he went boldly to Jupiter and begged him to give fire to men, that so they might have a little comfort through the long, dreary months of winter.

"Not a spark will I give," said Jupiter. "No, indeed! Why, if men had fire they might become strong and wise like ourselves, and after a while they would drive us out of our kingdom. It is best for them to be poor and ignorant, that so we Mighty Ones may thrive and be happy."

News

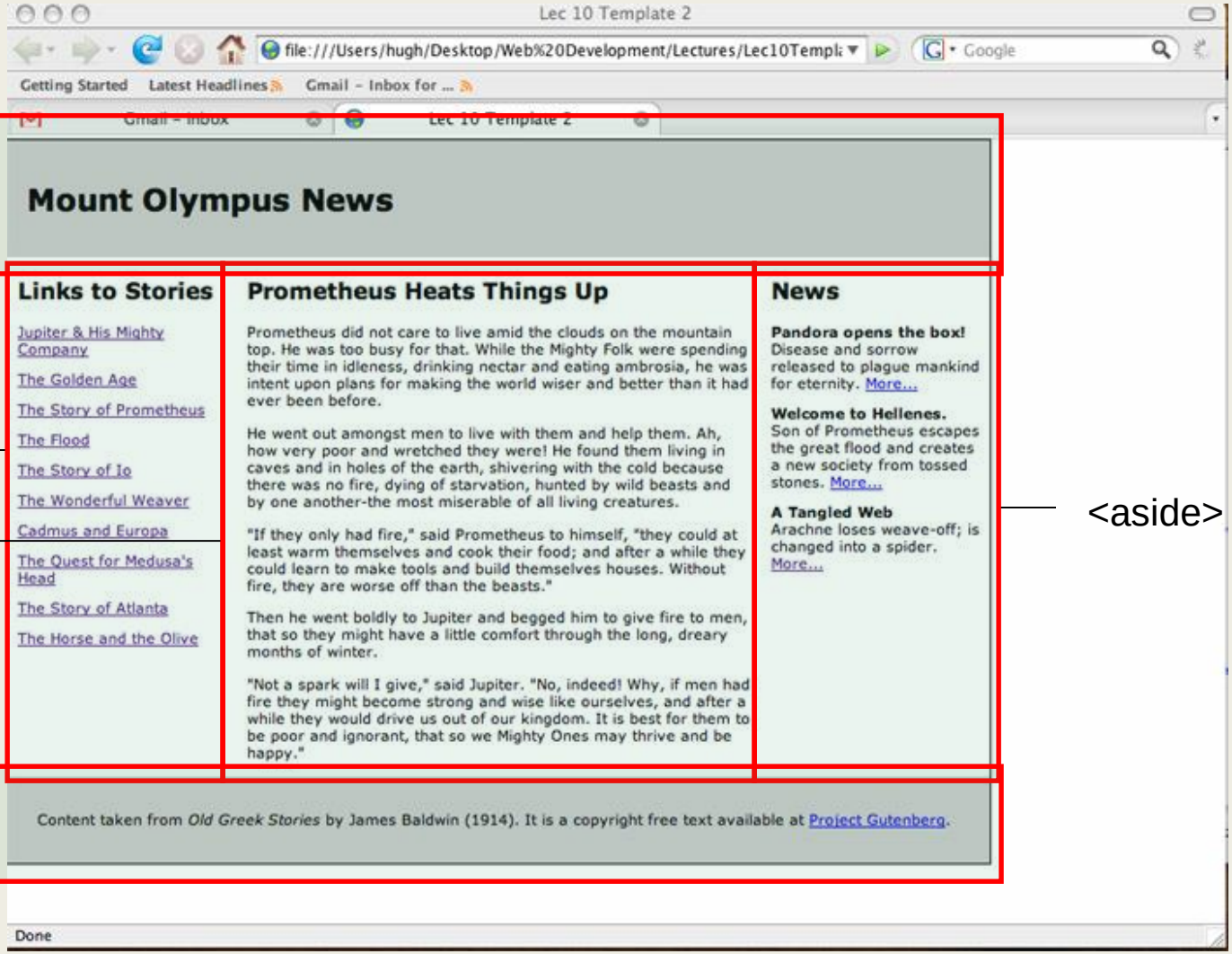
Pandora opens the box!
Disease and sorrow released to plague mankind for eternity. More...

Welcome to Hellenes.
Son of Prometheus escapes the great flood and creates a new society from tossed stones. More...

A Tangled Web
Arachne loses weave-off; is changed into a spider. More...

Content taken from *Old Greek Stories* by James Baldwin (1914). It is a copyright free text available at Project Gutenberg.

Multicolumn Layouts Using Floats



Multicolumn Layouts Using Floats

```
#container {
    width: 750px;
    border: solid 1px;
}

header {
    background: #CCC;
    padding: 15px;
    }

nav {
    float: left;
    width: 175px;
}
```

Entire page is of fixed width of 750 pixels

Border is optional …

Header appears first in source so will be placed at top of page

Left-hand column floated to left

Fixed at 175 pixels in width

Multicolumn Layouts Using Floats

```
section {
    float: left;
    width: 400px;
    }

aside {
    float: left;
    width: 175px;
}

footer {
    clear: both;
    padding: 15px;
    background: #CCC;
    }
```

Float *section* to the left also

Make it 400 pixels in width

Float *aside* to the left also. Third column.

Make it 175 pixels in width

Make sure footer starts below *all* floated elements

Multicolumn Layouts Using Floats

- Note that there is no padding, borders or margins on any of the floated elements.
- This is to ensure that the sum of their widths is exactly equal to the width of the container.
- 175 + 400 + 175 = 750.
- Space within each part of the webpage is achieved by applying padding to the elements it contains using the following style rule:

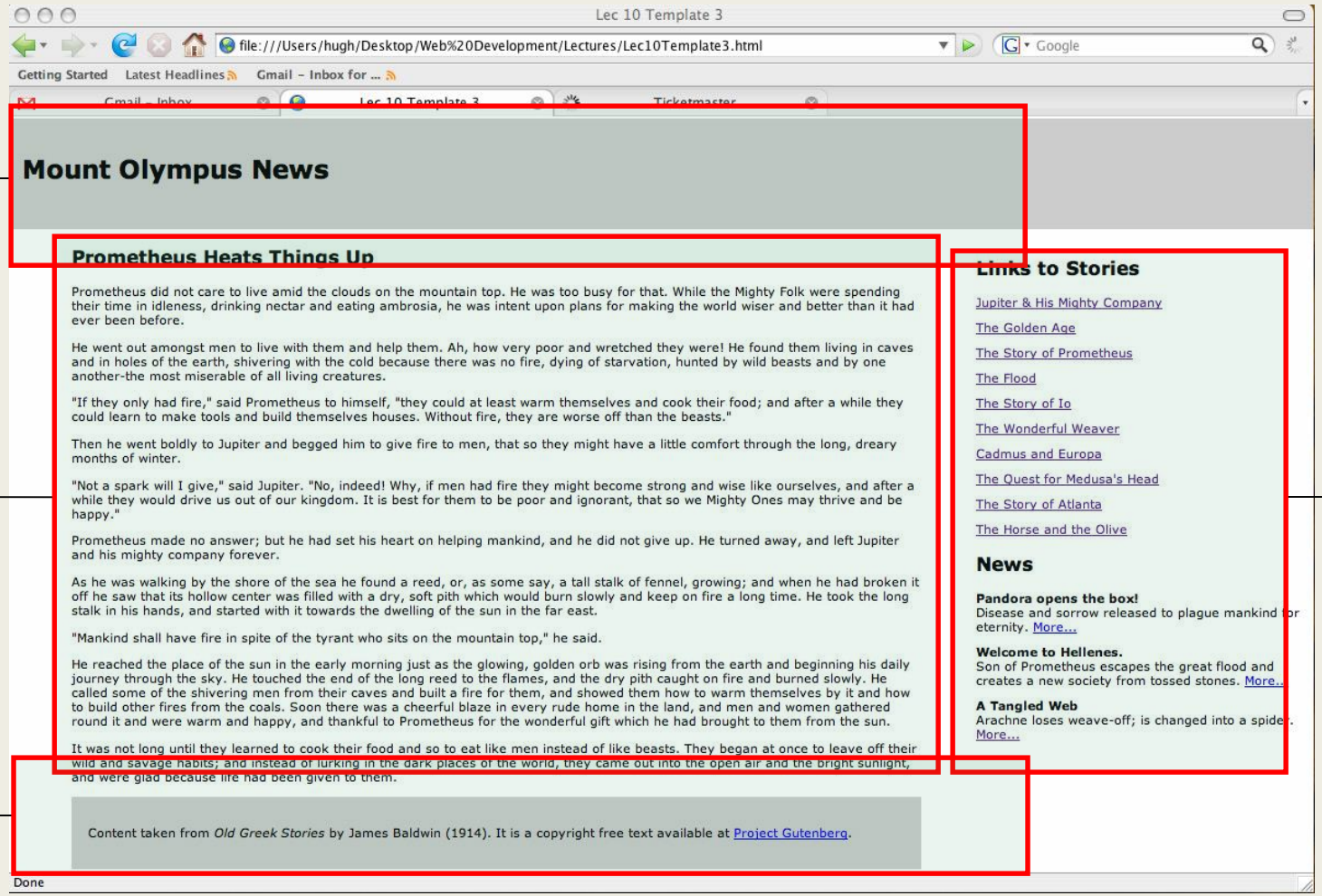
```
h2, ul, p { padding: 0px 8px; }
```

Multicolumn Layouts Using Floats

- **Template 3**
- Two Column with narrow footer
- Method: POSITIONED
- Layout: LIQUID

- This is similar to Template 1.
- The differences are that firstly it uses absolute positioning instead of floats, and secondly the footer does not extend all the way across the page.
- Markup is straightforward. It's just four elements - *header*, *section*, *aside* (right hand column) and *footer*.

Multicolumn Layouts Using Positioning



Multicolumn Layouts Using Floats

```
header {
      height: 70px;
      background: #CCC;
      padding: 15px;
      }


section {
      margin-right: 30%;
      margin-left: 5%;
}
aside {
      position: absolute;
      top: 100px;
      right: 0px;
      width: 25%;
      padding: 10px;
}
```

***header* is fixed at a height of 70 pixels**
Total height of header is 15+70+15 = 100

***section* div has a right-margin wide enough to make space for the <*aside*>**

***aside* div is positioned exactly 100 pixels from the top, and right up against the right side of the page.**
Making it 25% the width of the window ensures it will fit inside the 30% right margin of the *main* div.

Multicolumn Layouts Using Floats

```
footer {
    margin-right: 30%;
    margin-left: 5%;
    padding: 15px;
    background: #CCC;
    }
```

***footer* has the same margin as <*section*> it slots under it correctly.**

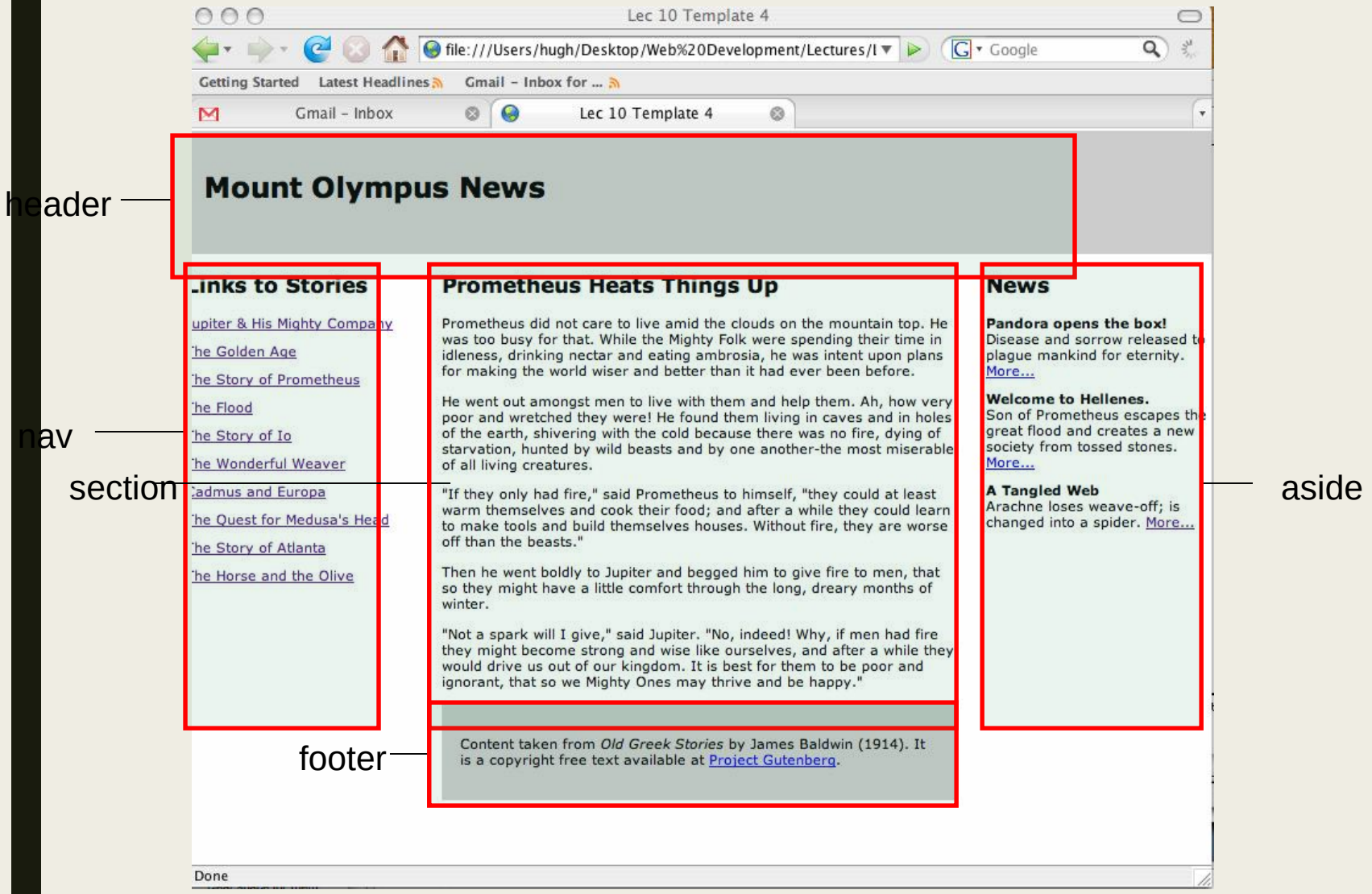
- Giving a 30% right-margin to *footer* also ensures that it will not become overlapped by a very long *<aside>* column (the right-hand one).

Multicolumn Layouts Using Floats

- **Template 4**
- Three Column with narrow footer
- Method: POSITIONED
- Layout: LIQUID

- In this template both the left and right columns are absolutely positioned.
- Margins are then applied to both sides of the middle main content column to clear space for them.

Multicolumn Layouts Using Floats



Multicolumn Layouts Using Floats

```
header {
      height: 70px;
      background: #CCC;
      padding: 15px;
      }

section {
      margin-right: 25%;
      margin-left: 25%;
}

nav {
      position: absolute;
      top: 100px;
      left: 0px;
      width: 22%;
}
```

header **is fixed at a height of 70 pixels**
Total height of header is 15+70+15 = 100

section **has a right-margin wide enough to make space for the *news* div and a left**
margin big enough to fit the *links* div.

nav **is positioned exactly 100 pixels from the top, and right up against the left side of the page.**
Making it 22% the width of the window ensures it will fit nicely inside the 25% right margin of the *main* div.

Multicolumn Layouts Using Floats

```
aside {
    position: absolute;
    top: 100px;
    right: 0px;
    width: 22%;
}



footer {
    margin-right: 25%;
    margin-left: 25%;
    padding: 15px;
    background: #CCC;
    }
```

Right hand <*aside*> column done in a similar manner to the left hand *links* one.

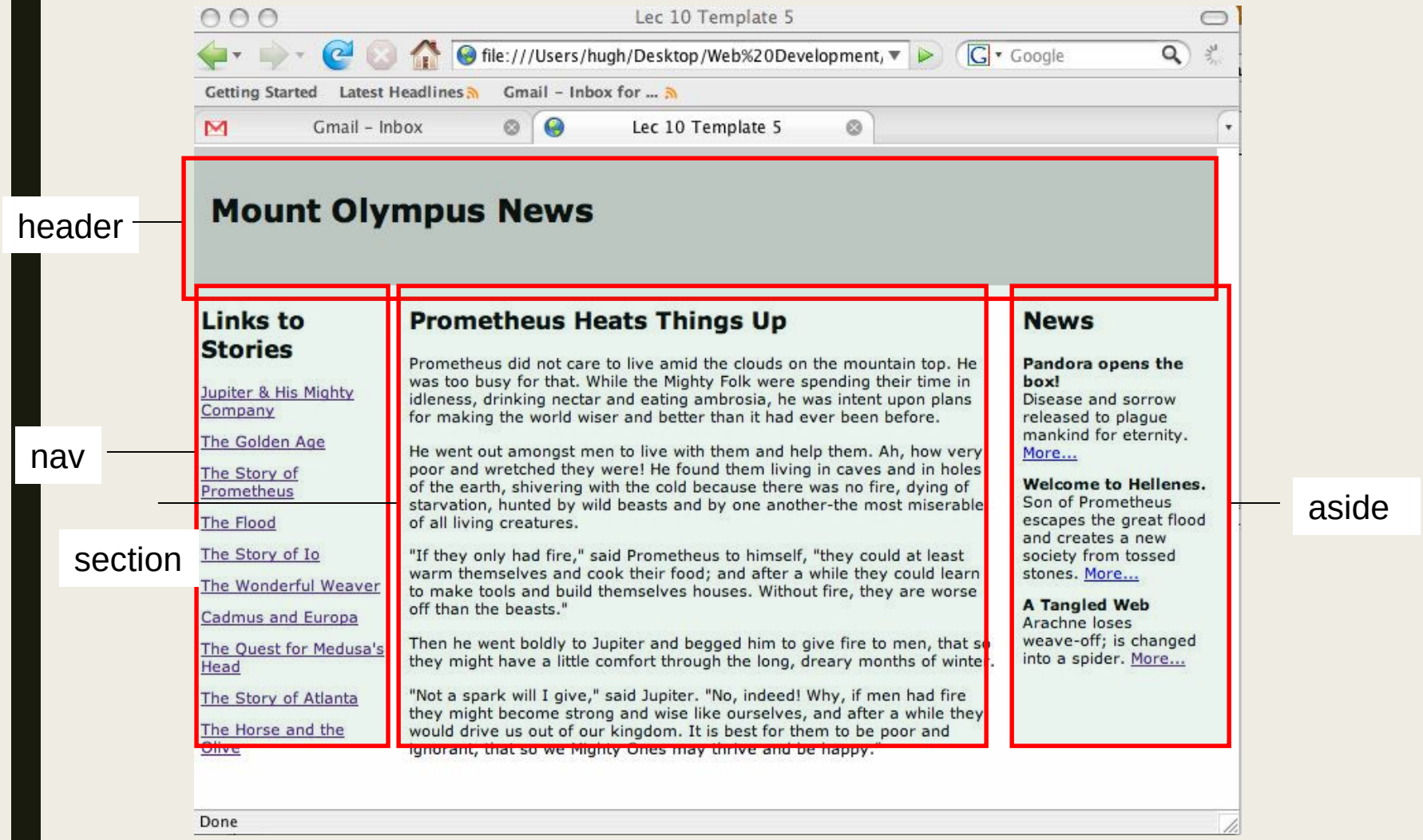
***footer* is done the same way as in the last template. 25% is put on both sides so that it fit exactly under the *main* div.**

Multicolumn Layouts Using Floats

- **Template 5**
- Three Column with padding between columns
- Method: POSITIONED
- Layout: FIXED

- Finally let's look at how to do a three column fixed layout using positioning.
- Everything here is precisely positioned so nothing will resize with the browser window.
- No footer in this template!

Multicolumn Layouts Using Floats



Centering a Fixed Width Page

- All the templates have been left-aligned.
- Suppose we want one centred?
- There are two ways to do this.
- Firstly, the official CSS way!
- This says that what you do is specify a width for the container **div** element and then set it's left and right margins to auto.

```
container {
   position:relative;
   width: 750px;
   margin-right: auto;
   margin-left: auto; }
```

Lecture Summary

- Width property
- Float property
- Clear property
- Floating text, images, first letter
- Creating sidebars, links, multi-column page layouts
- Making Floating elements fit
- Problem of floating too late
- Fixed/Liquid