WEB DEVELOPMENT

Lecture 01
Mark-Up Languages

What is a Web Server?

http://www.bpc.ac.uk/staff/kevin.htm

Web Server

```
GET  /staff/kevin.htm  HTTP/1.1
Host: www.bpc.ac.uk
```

Path
translation

Internet

Kevin.htm

```
<html>
 <body>
   Hello world
 </body>
</html>
```

`C:\inetpub\wwwroot\staff\kevin.htm`
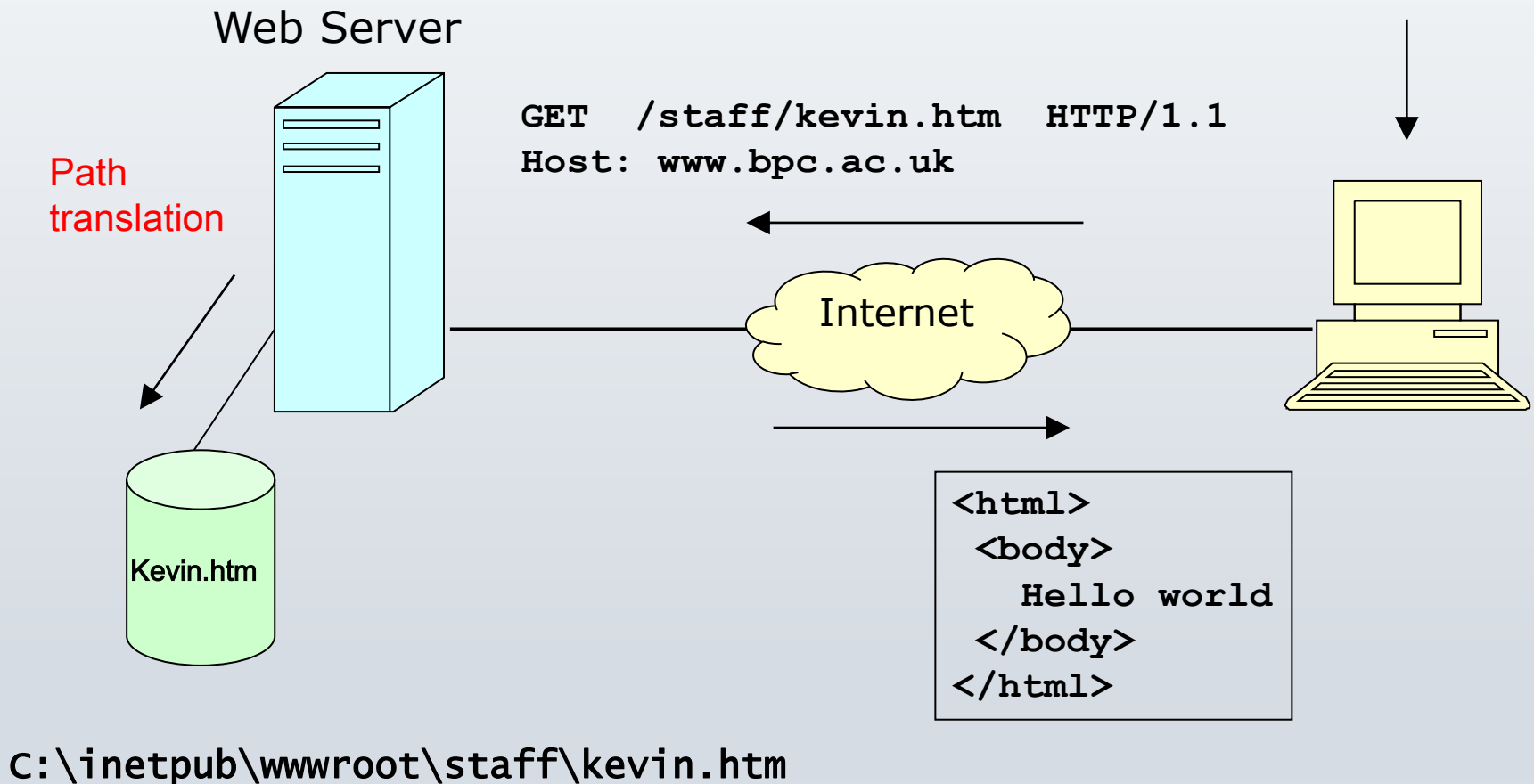
What is HTML?

Webpages are mostly written in HTML.

This stands for:

Hyper
Text
Markup
Language

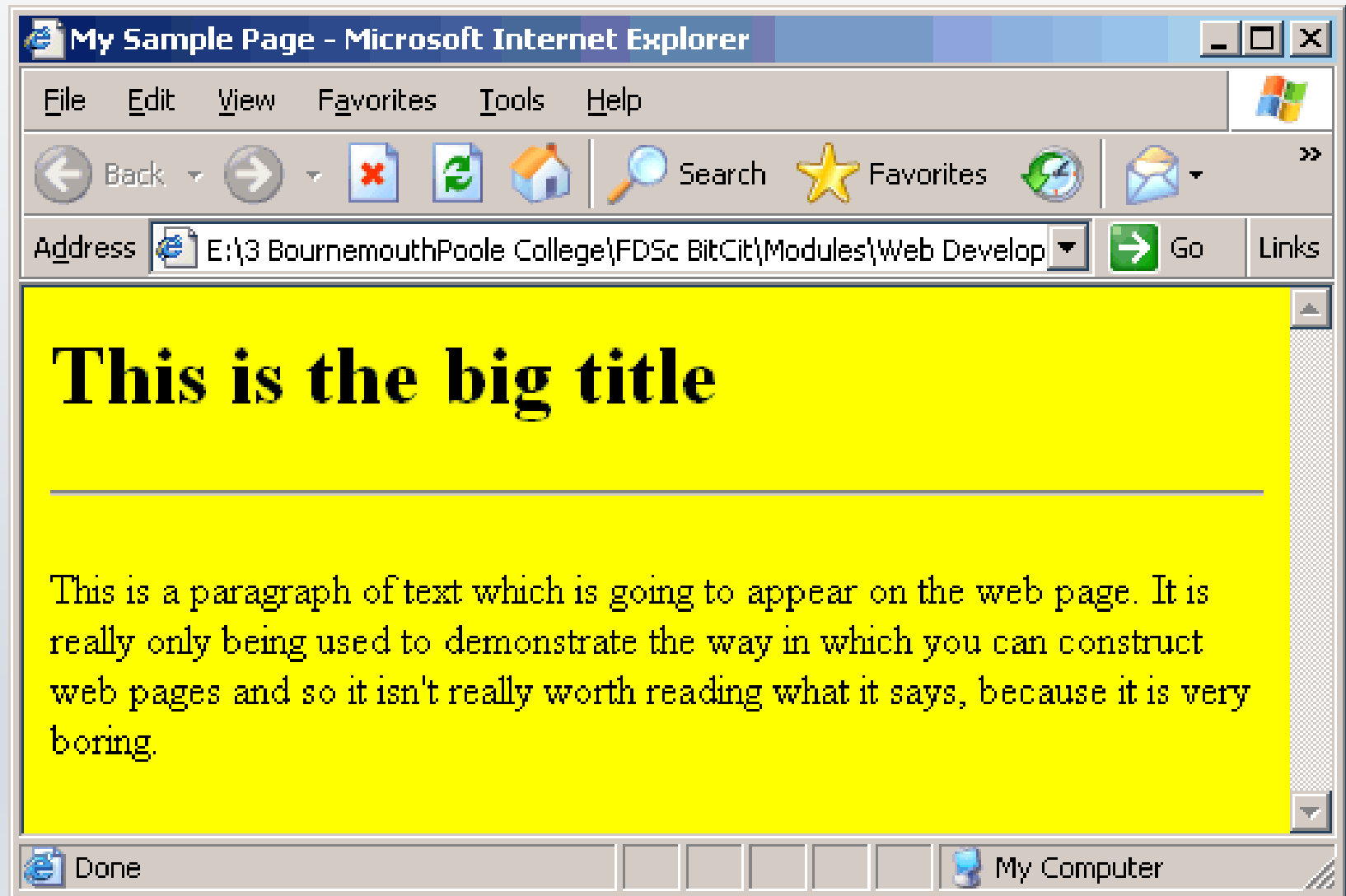# Example of HTML

```
<html>
  <head>
    <title>My Sample Page</title>
  </head>
  <body bgcolor="yellow">
    <h1>This is the big title</h1>
    <hr>
    <p>This is a paragraph of text which is going to
appear on the web page.  It is really only being used
to demonstrate the way in which you can construct web
pages and so it isn't really worth reading what it
says, because it is very boring.</p>
  </body>
</html>
```

When displayed in a browser….

Note that….

The markup consists of a set of tags, or elements.

`<title>`   `<p>`   `<h1>`

Each opening tag has a closing tag

`<html>`   `</html>`   `<body> </body>`

Some tags have attributes, which can be given values.

`<body bgcolor="yellow">`

bgcolor is actually an out of date, deprecated attribute, which is no longer used.

What is Hypertext?

"Hypertext is text which is not constrained to be linear.

Hypertext is text which contains links to other texts. The term was coined by Ted Nelson around 1965.

HyperMedia is a term used for hypertext which is not constrained to be text: it can include graphics, video and sound , for example. Apparently Ted Nelson was the first to use this term too.

Hypertext and HyperMedia are concepts, not products."

`Taken from www.w3.org/whatis.html – (9/8/07)`

What is a Markup Language?

Printers used to mark up a paper manuscript with symbols to indicate what typeface, size and style should be used to layout the page.
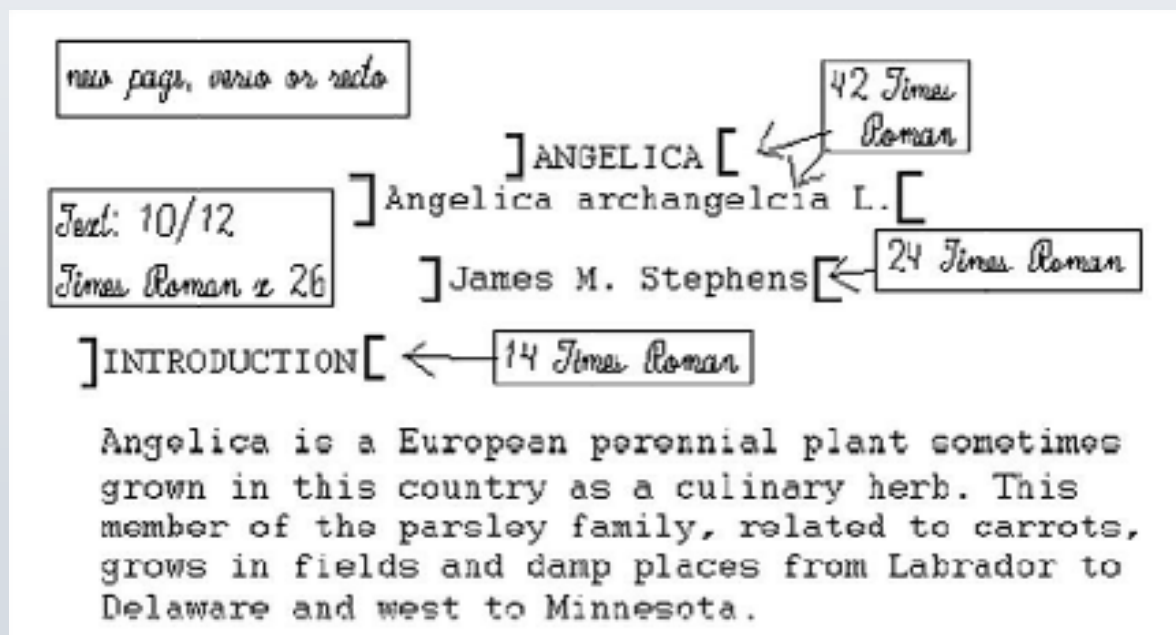
What is a Markup Language?

Standardised General Markup Language (SGML) developed by IBM in the 1960s as a way of encoding data. Used for

- Boeing Aircraft Technical Documentation
- Oxford English Dictionary (2nd edition)

Very complex set of tags and rules, which are laid out in a Document Type Definition (DTD).

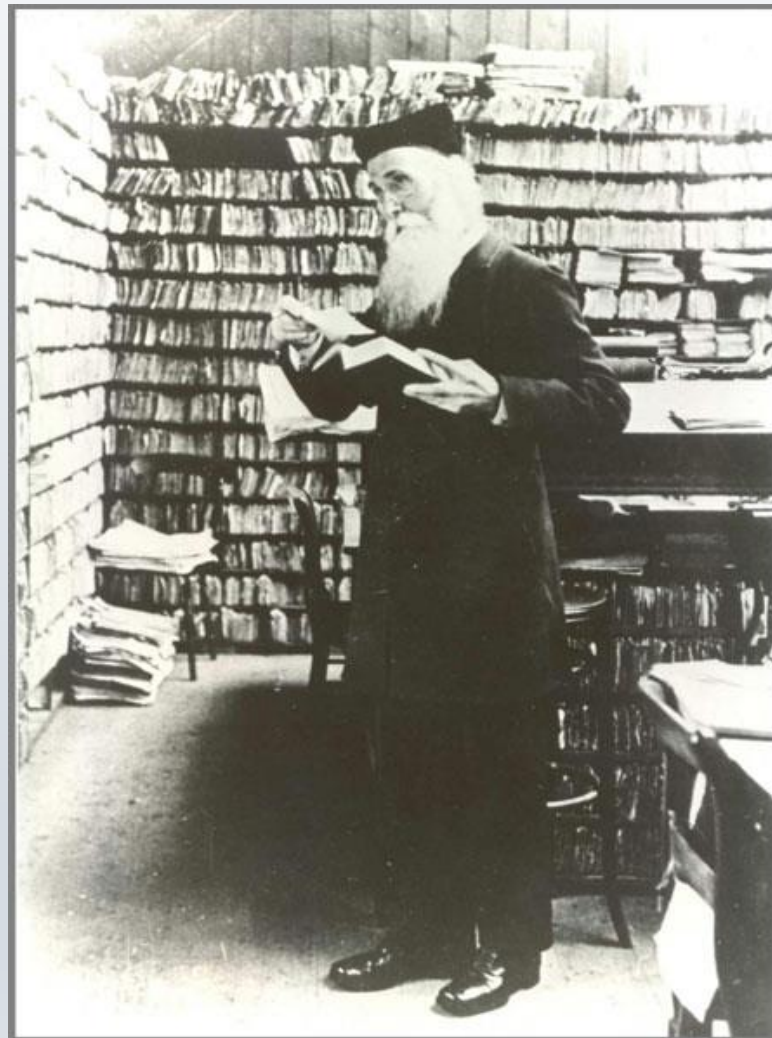# Oxford English Dictionary

First edition started 1857

They thought it would take 10 years to finish.

By 1888 they had not got very far.

First edition was fully published in 1933.

By 1950, they realised that they needed to start work on the second edition.

Example of an OED entry in SGML


```
Document: Bungler OED                    At: "<entry>"

<entry>
  <hwsec>
    <hwgp>
      <hwlem>bungler</hwlem>
      <pron>b<I>ʌ</I>ŋgləɹ</pron>. </hwgp>
    <vfl>Also <vd>6</vd> <vf>bongler</vf>,
      </vfl>
    <etym>f. as prec. + <xra><xlem>-ER</xle

  <sen>One who bungles; a clumsy unskilful
    <quot>
      <qdat>1533 </qdat>
      <auth>MORE </auth>
      <wk>Answ. Poyson. Bk. </wk>Wks. (1557
      <qtxt>He is euen but a very bungler.
```

Taken from: http://en.wikipedia.org/wiki/Sgml – (16/8/07)

History of HTML

1980s   Computers are all text-based, with a command line interface.

1990   Tim Berners-Lee develops an in-house document sharing system for use at CERN. It includes the first version of:

- HTML (based on SGML)

- HTTP, a protocol for transmitting HTML documents

- A GUI browser which he calls 'WorldWideWeb'.

Screenshot from Berners-Lee's Computer
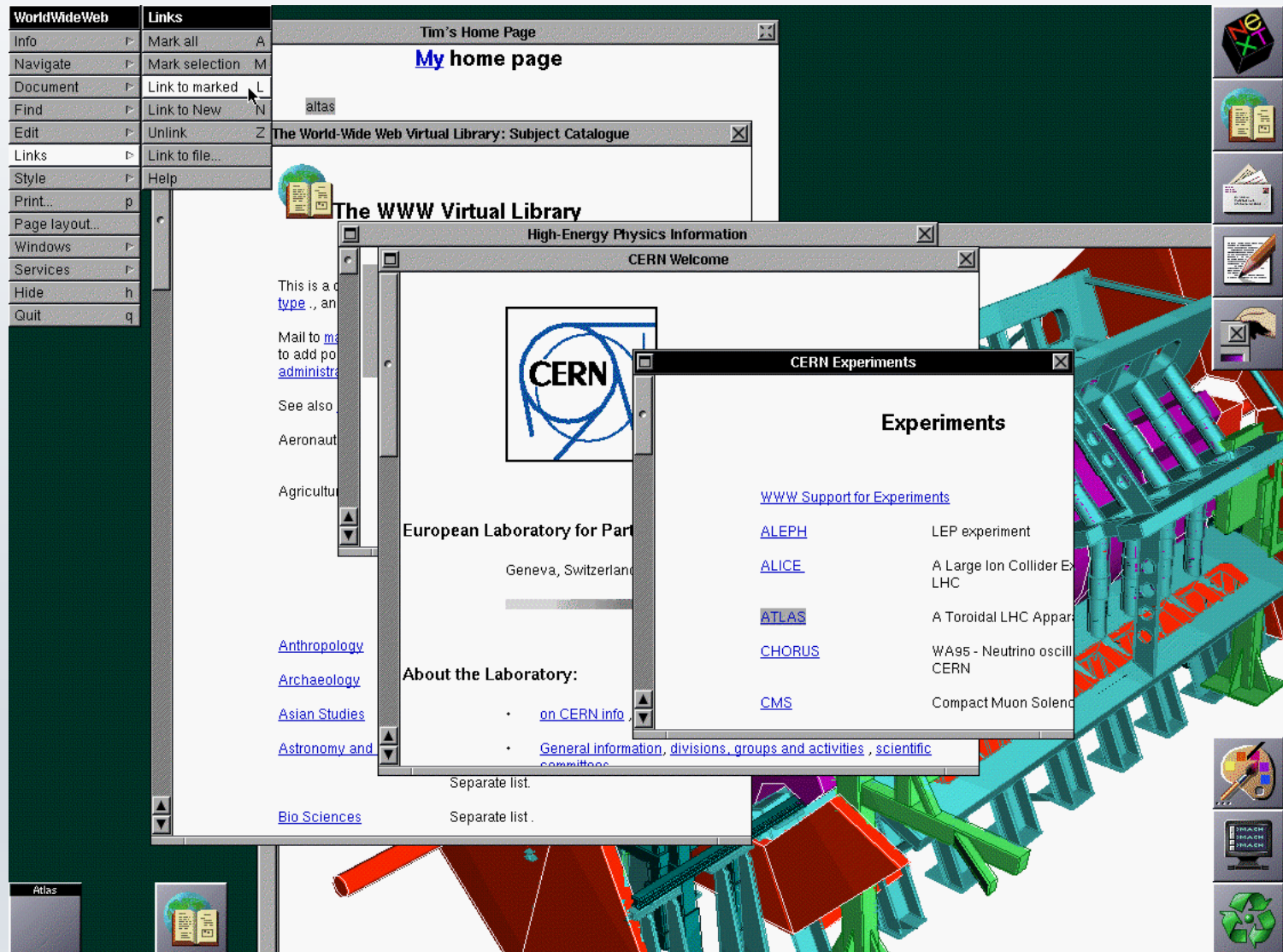
1993

Mosaic, a GUI web browser, written by Marc Andreessen and Eric Bina at NCSA, spreads like wildfire through the global academic community.

1994

Tim Berners-Lee leaves CERN and sets up the World Wide Web Consortium (W3C).

An organisation which starts trying to develop standards for web based technologies.

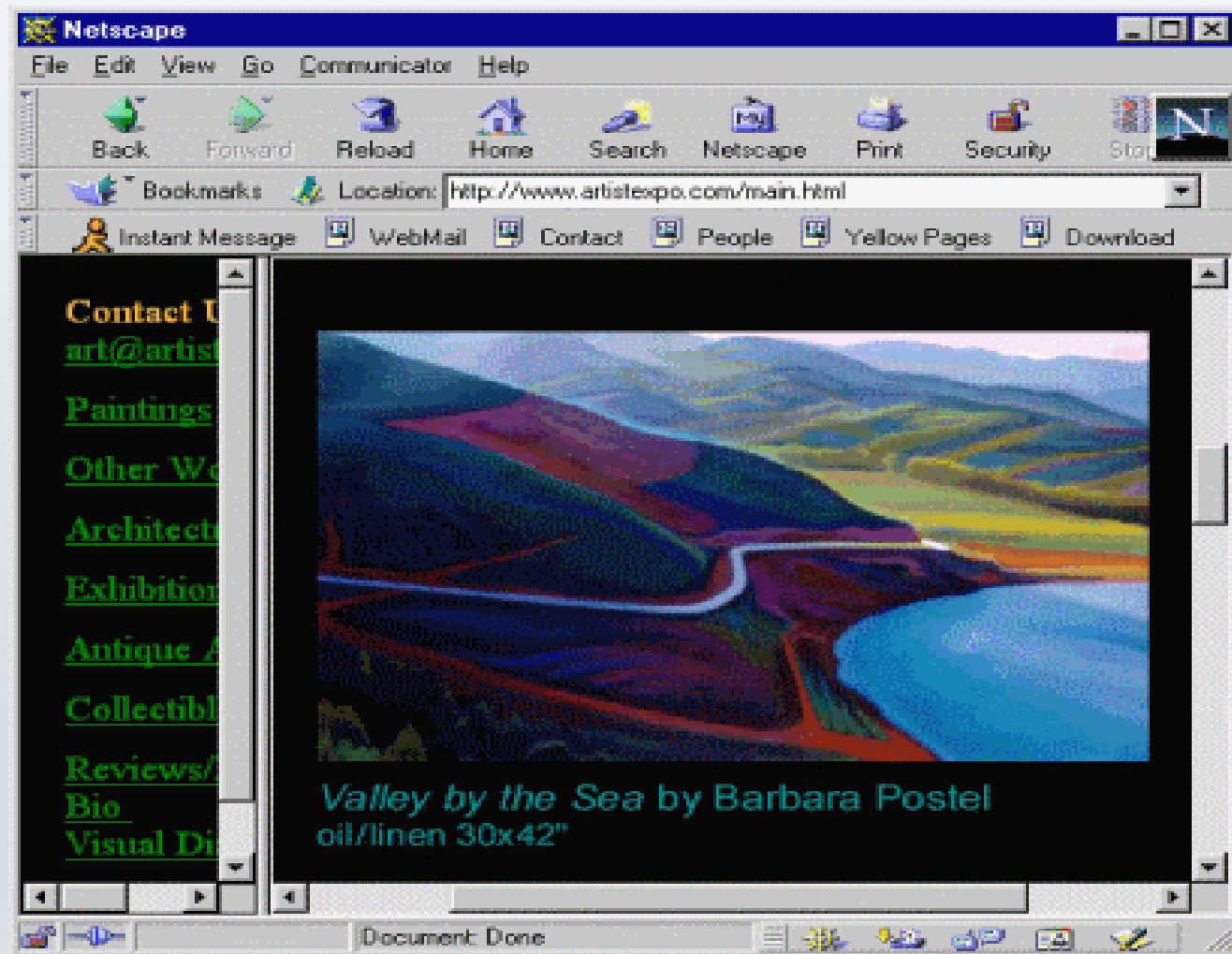But this takes time. It is not until:

1995   HTML v3.0
1996   CSS v1

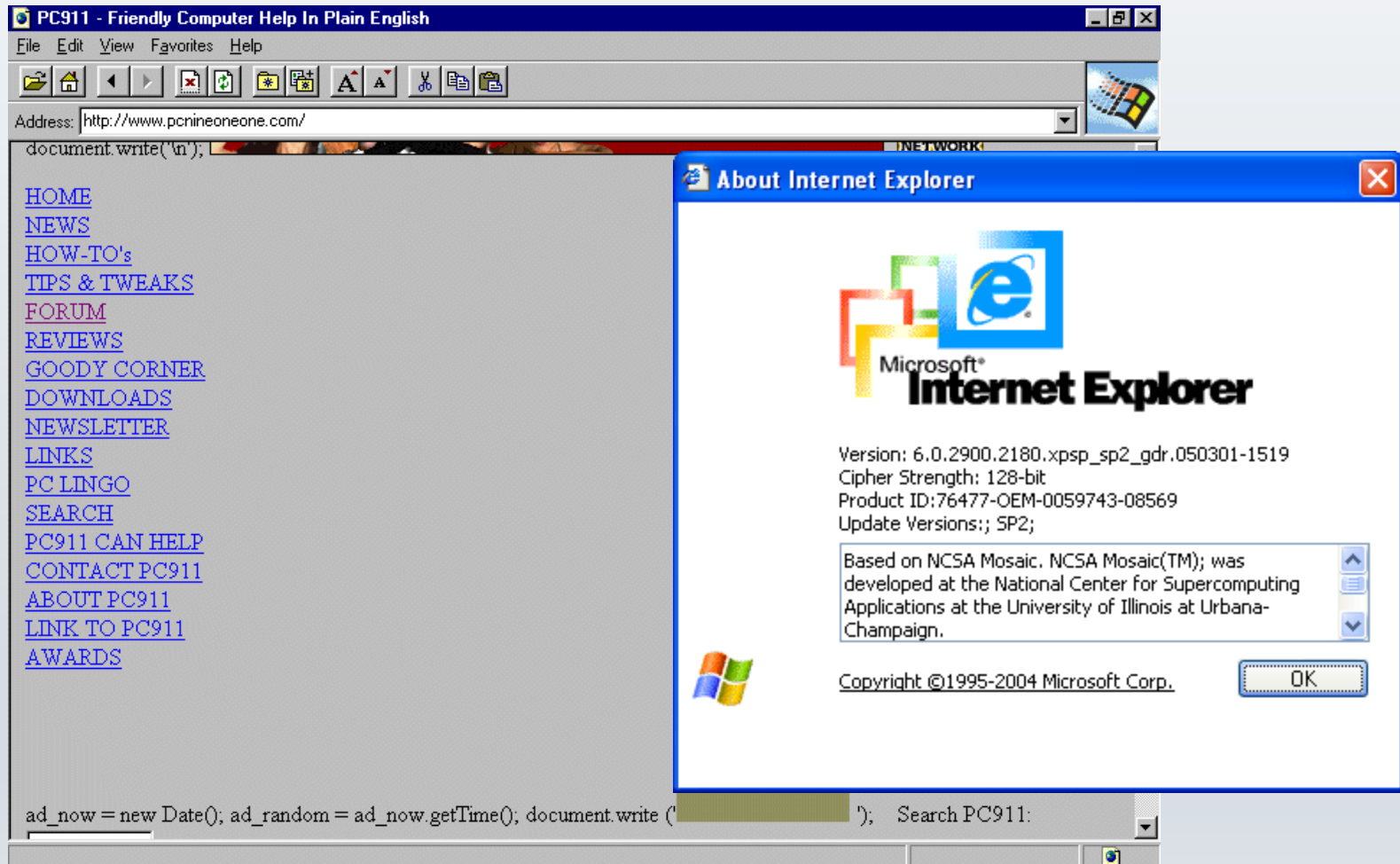are published.

In the meantime….

1994

Andreessen forms the company 'Netscape' which produces a browser called 'Mozilla', later called 'Netscape Navigator'.

1995

Microsoft produces Internet Explorer v1 and ships it with Windows 95.

Internet use enters the mainstream.

The Browser Wars   1995 - 2000

Microsoft and Netscape fight for market share, trying to make sure their browser is the most widely used. As a result:

- They tend to add new features instead of fixing bugs.
- They add proprietry features instead of obeying standards.

HTML development is fairly chaotic during these five years.

- There are lots of different versions of HTML around.
- They are not always compatible.
- The browser manufacturers try to make HTML do things that it was never meant to do.

Microsoft wins and in 2000 Netscape is sold off to AOL.

Bad HTML

```
<HTML>
  <HEAD>
    <TITLE>My Sample Page</TITLE>
  </HEAD>
  <BODY BGCOLOR=#FFCC00>
    <P>
      <H1><FONT COLOR=RED>
        <U>These are physical styles</FONT></U>
      </H1><BR>
      <TT>This is in typewriter style.</TT><BR>
      <BLINK><I>This is in italic.</I></BLINK><BR>
      <STRIKE>This is in strikeout style.</STRIKE>
        <BR>
      <B><FONT SIZE=20>This is in boldface.</FONT></B>
        <BR>
      <U>This is in underline style</U><BR>
    </P>
  </BODY>
</HTML>
```

Which looks like this:
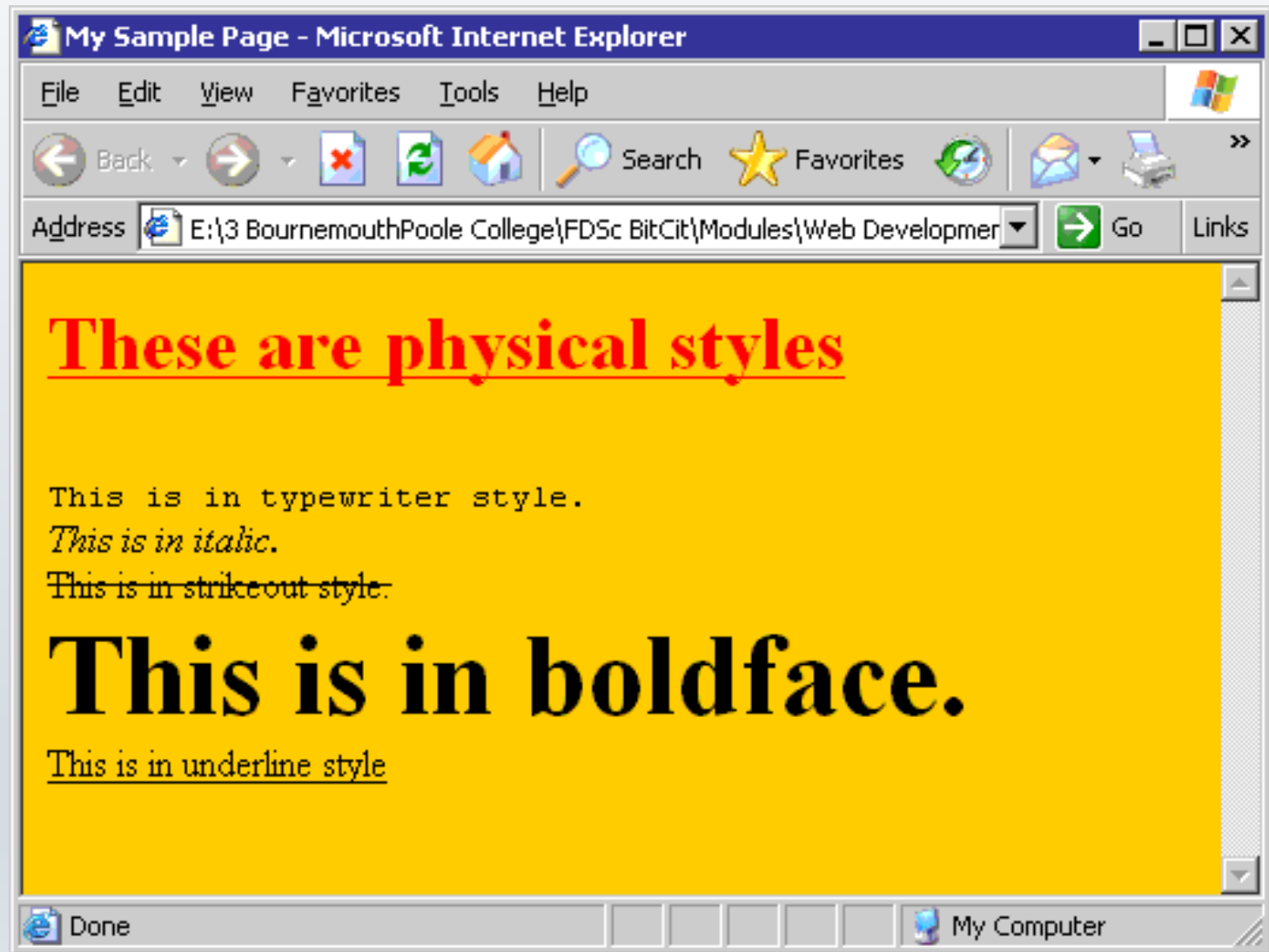
Constructing a Webpage
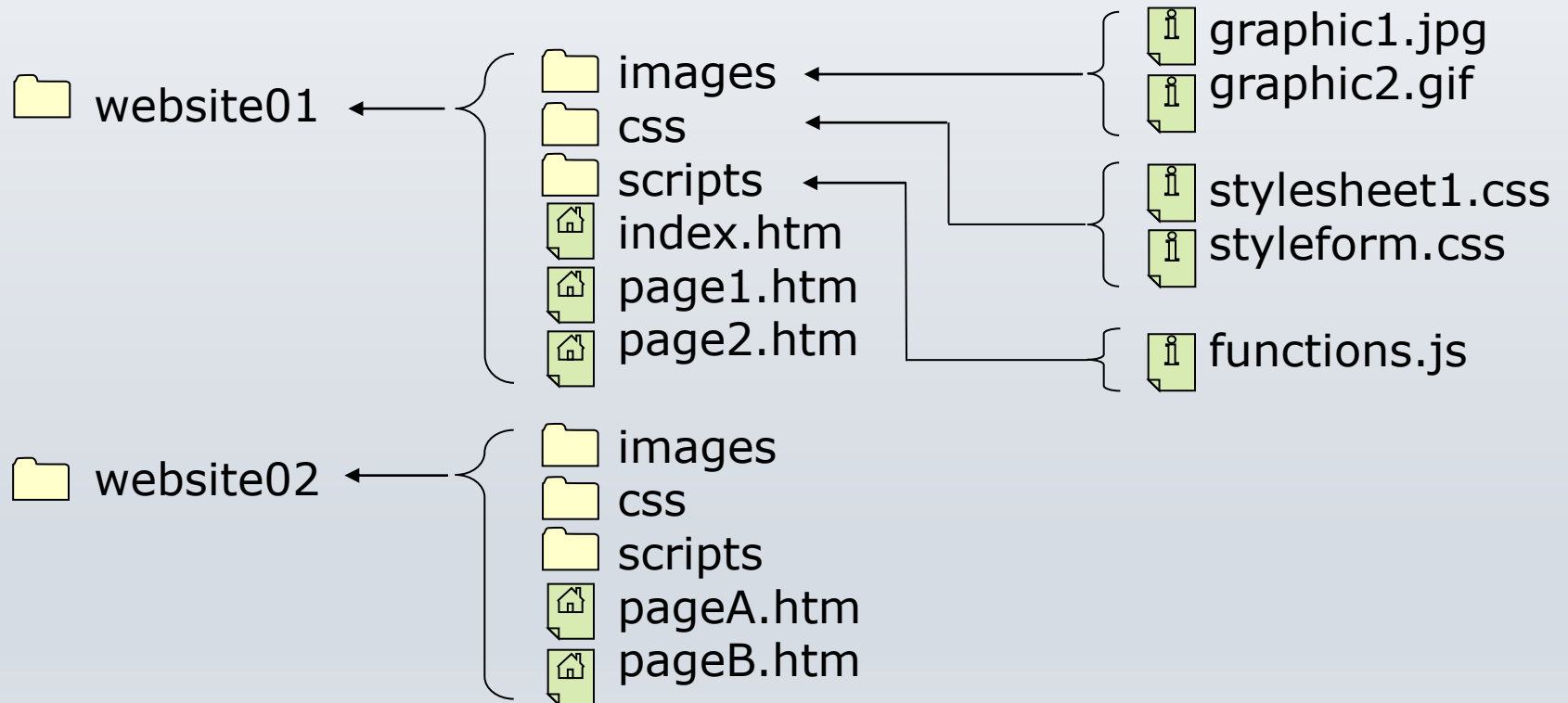
The Structural Layer
- **What it is**
  - XHTML

The Presentation Layer
- **What it looks like**
  - Cascading Style Sheets (CSS)

The Behavioural Layer
- **What it does**
  - JavaScript (or similar scripting language)

Keep Them Separate

A well designed website will keep the three types of code in separate files – or even separate folders.

📁 website01 →
- 📁 images →
  - 🖼 graphic1.jpg
  - 🖼 graphic2.gif
- 📁 css →
  - 📄 stylesheet1.css
  - 📄 styleform.css
- 📁 scripts →
  - 📄 functions.js
- 🏠 index.htm
- 🏠 page1.htm
- 🏠 page2.htm

📁 website02 →
- 📁 images
- 📁 css
- 📁 scripts
- 🏠 pageA.htm
- 🏠 pageB.htm

HTML vs XHTML

Most browsers will allow you to write lazy HTML.

You can:

- Use tags that control how the document looks

- Mix upper and lower case element names.

- Leave quotes off attribute values.

- Not bother with some closing tags.

- Leave off some attribute values.

- Not bother to nest all elements correctly.

In XHTML you must always…

- Use lower case for all element and attribute names.

- Put quotes around attribute values (even single words and numbers).

- Terminate elements….

- …Even empty elements. (
  or
</br>)

- Use explicit attribute values (checked="checked").

- Make sure that all elements are nested correctly.

- Use the id attribute in preference the name attribute.

- Place CSS and Scripts in their own files.

XHTML is a more rigorous, precise version of HTML.