

Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

M150 Data, Computing and Information

Differences between data and information

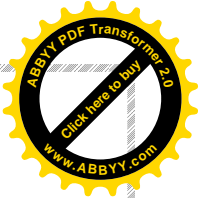
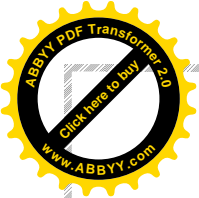
The interchange of the words data and information is widespread, but M150 should help you to develop a clearer understanding of the differences between the two.

Data

- Facts, statistics used for reference or analysis.
- Numbers, characters, symbols, images etc., which can be processed by a computer.
- Data must be interpreted, by a human or machine, to derive meaning
- "Data is a representation of information" *
- Latin 'datum' meaning "that which is given"
- Data plural, datum singular (M150 adopts the general use of data as singular.

Information

- Knowledge derived from study, experience (by the senses), or instruction.
- Communication of intelligence.
- "Information is any kind of knowledge that is exchangeable amongst people, about things, facts, concepts, etc., in some context." *
- "Information is interpreted data" *



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

Data or Datum?

Unit 1 p10 states in a sidebar:

"The word 'data' is a plural Latin word but it is generally used as a singular word in English."

Here are some definitions:

data n, pl. 1 Facts on which a decision is to be based. 2 Facts to be processed by computer.

datum n. (pl. data) item of data"

Reference:

Oxford Quick Reference Dictionary and Thesaurus - Oxford University Press

data

noun 1 facts and statistics used for reference or analysis. 2 the quantities, characters, or symbols on which operations are performed by a computer.

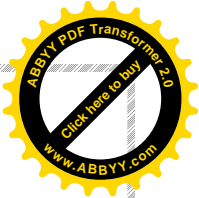
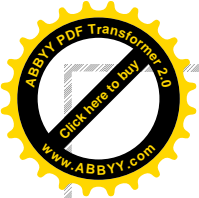
USAGE Traditionally and in technical use data is treated as a plural, as in Latin it is the plural of datum. In modern non-scientific use, however, it is often treated as a singular, and sentences such as data was collected over a number of years are now acceptable.

ORIGIN Latin, plural of DATUM.

datum

noun (pl. data) 1 a piece of information. 2 an assumption or premise from which inferences may be drawn. 3 a fixed starting point of a scale or operation.

ORIGIN Latin, 'something given'."



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

Inevitably some will disagree with the singular/plural usage and with the precise meaning of data. For example, you might think that most would agree with a measurement being a form of data.

To initiate an online bunfight (sometimes referred to as an "academic discussion"), follow these instructions:

1. Choose a FirstClass conference (your regional M150 conference will probably suffice).
2. Pick an example from a course unit (try sticking a pin in any page) e.g. Unit 1 p5, 1.1 "data (such as a person's name, or the number of a bus route)".
3. Copy it into your message.
4. Write, "The use of data in this example as opposed to datum is incorrect and betrays a total disregard for the English language. It should read datum. Data is plural, not singular!"
5. Dig a large hole in the ground.
6. Climb in and wait for the response.

Here is a part of an online discussion taken from a FirstClass conference. Please note that all correspondents follow good Netiquette. If you are unfamiliar with this term, read this [article](#) - "Netiquette, or the social conventions of computer conferencing".

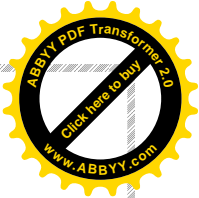
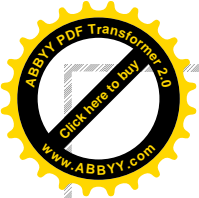
Correspondent 1 writes:

"I (and many other people I know) use datum and data correctly as the one word that they are (is?) in singular and plural form. All my documentation is written with expressions like 'the data are stored as . . .', and so on.

It is pleasing to see that many authors (including some from the USA) correctly identify data as a plural."

Correspondent 2 replies to Correspondent 1:

"... I have no problem with the use of data as you refer to it but, in engineering, datum is the reference point or line with which all measurements are compared. It is also used on maps e.g. the datum against which heights are measured is sea level. The source of the two words is common but the meanings are different."



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

Correspondent 1 replies to Correspondent 2:

"Sorry if I misunderstood. I think the engineering meaning (with which I am cognizant) is really just a slightly special adaptation of the general case: datum means literally 'given thing'.

Thus data are 'given things', and one distinction that I always try to maintain professionally is between data (given things, like Planck's constant or a set of engineering datums - sorry, I have to express it like that here!), and measured things, such as experimental readings.

Many people refer to measurements as data, but this is not correct as they are not really given. Except of course, that someone, say A, measures something and passes the measurement on to B, so to B it is an item of data (that is a datum). (So it all begins to look a bit arbitrary!)."

Correspondent 3 replies to Correspondent 1:

"Your explanation of the origin of the word datum is interesting. However, I've always used the word parameter to signify given values such as fixed constants, and data to imply that a measurement has been made. This may not be linguistically accurate, but I think it's common usage."

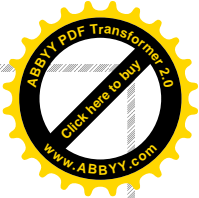
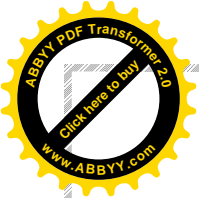
Correspondent 4 replies to Correspondent 3:

"Your usage and (Correspondent1's) are not mutually exclusive. A datum is not a parameter but a point from which measurements are made."

Transforming Data into Information

If you go you will find a form that I'm going to ask you to process. When you finish return to this page.

OK, hopefully, you entered some data, clicked the Submit button and were presented with another web page that displayed that data. If you didn't fill in the form as I wished, you were prompted to try again.



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

Now try this. Again come back when you have finished.

The two processes that you have just engaged with encompass the three main themes of M150.

1. You were presented with a web page and asked to enter certain values. In other words **data** was collected.
2. You pressed the submit button. At which point that data was stored and operated on by a program. The **program**, written in JavaScript, was embedded in the webpage and manipulated the data before displaying some text, marked up in HTML, on another webpage.
3. In the case of the first form this simply reflected the values you entered. With the second form, the data was operated on in such a way as to display some **information** about you, i.e. whether or not you are new to the OU.

These exercises encapsulate a major theme of the course, namely the **transformation of data into information**.

Binary Number System

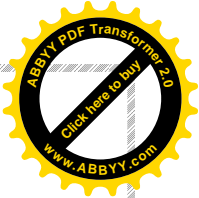
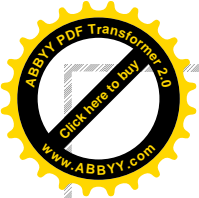
Before looking at the binary representation of numbers, it is worth taking a quick look at the representation and meaning of decimal, or denary, numbers.

This system employs a base 10 representation. Another way of describing this is to say that it has a radix of 10, meaning that each digit to the left of the decimal point represents increasing powers of 10 and digits to the right of the point represent decreasing powers of 10.

Consider the decimal number 251, which can be illustrated as follows:

$10^2=100$	$10^1=100$	$10^0=100$
2	5	1

In other words, 251 represents the addition of $2 \times 100 + 5 \times 10 + 1 \times 1$.



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

Binary representation (radix 2) simply offers another form of numeric representation by adding increasing powers of 2, rather than 10.

The decimal value 251 can be represented by the binary representation 11111011.

$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
1	1	1	1	1	0	1	1

Computer programs operate on data and ultimately enable humans to derive meaning from these representations.

The binary number system is the simplest of all number systems. Its representation requires a combination of just two numbers, 0 and 1, to represent any binary value. 1s and 0s are binary digits or bits. An 8-bit unit is called a byte; employed by most computers as the basic unit of data representation. A byte is a subdivision of a word.

A word is "a vector of bits that is treated as a unit by the computer hardware. The number of bits, referred to as the word length or word size, is now usually 16 or 32. The memory of a computer is divided into words (and possibly subdivided into bytes). A word is usually long enough to contain an instruction or an integer."

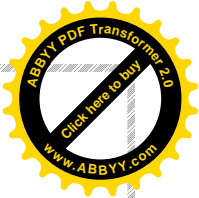
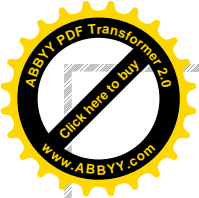
"word" A Dictionary of Computing. Oxford University Press, 1996. Oxford Reference Online. Oxford University Press. Open University. 6 March 2004 <<http://www.oxfordreference.com/views/ENTRY.html?subview=Main&entry=t11.e5841>>

How does a computer utilize the binary system?

At its lowest level (machine level) a computer operates by switching on or off millions of transistors at great speed. These transistors, a form of switch, are etched onto the microprocessor/chip.

A computer processes data, held in memory and represented in a binary form or code. Switches that are either "on" or "off" implement this.

A transistor/switch is a two-state device and can be in one of two states:



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

○ On, which represents the binary digit 1

○ Off, which represents the binary digit 0

Consequently, 8 transistors with appropriate state can, for example, represent the binary code 11111011.

On	On	On	On	On	Off	On	On
----	----	----	----	----	-----	----	----

Example Usage

IP (Internet Protocol) addresses are numeric identifiers and are assigned to any machine on an IP network. They are software addresses and represent encoded information that enables the routing of datagrams, or packets, on the Internet. They consist of four bytes (32 bits). Each byte of an IP address, known as an octet, can take any value with a decimal equivalent of between 0 and 255. Here is a binary value for the IP address of BBC

<http://www.bbc.com>)

11010100 00111010 11111011 11000011

The decimal equivalent, which is far easier for humans to read, is

212.58.251.195

An IP address is written in dotted quad form and in the case of the BBC example, we can write 212.58.251.195

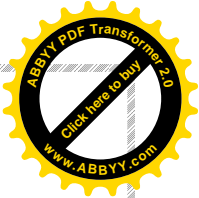
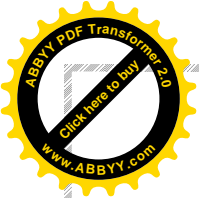
By entering <http://212.58.251.195> into a browser, a user can access the BBC website.

Note that when you enter <http://www.bbc.com> into a browser this name is resolved into a number by another computer on the Internet known as a Domain Name Server.

32 bits is $2^{32}-1$ or 4294967295 or 4.294967295×10^9 (approx. 4.3 billion, lots of potential addresses, although not enough to satisfy anticipated demand).

Octal Number System

Octal notation offers another means of representing numeric values. Unlike decimal, it is based on 8, or radix of 8, rather than 10, i.e. increasing powers of 8. Octal numbers can be represented by using a combination of 0,



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

1, 2, 3, 4, 5, 6, 7. For example, the decimal number 1209 is represented in the octal 2271

$$= 2 \times 512 + 2 \times 64 + 7 \times 8 + 1 \times 1$$

$8^3=512$	$8^2=64$	$8^1=8$	$8^0=1$
2	2	7	1

$$2 \times 512 = 1024 + 2 \times 64 = 128 + 7 \times 8 = 56 + 1 \times 1 = 1$$

An octal digit (0 to 7) can represent 3 bits (000 to 111). Consequently it can be regarded as a shorthand version of binary.

Octal notation was widely used to provide a more readable view of binary data. It has been mostly replaced by hexadecimal notation.

Hexadecimal Number System

A base 16 (radix 16) number system, i.e. increasing powers of 16.

Introduced by IBM - legend has it that the company refused to countenance the more accurate sexadecimal name.

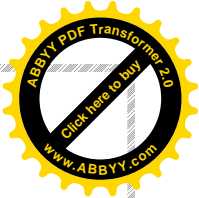
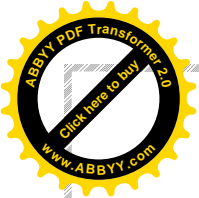
Besides the numbers 0 ..9, hex uses the letters A, B, C, D, E, and F to represent the decimal equivalents 10, 11, 12, 13, 14 and 15 ([see rules](#)).

For example, decimal value 31 has hex value 1F

$16^3=4096$	$16^2=256$	$16^1=16$	$16^0=1$
0	0	1	F

$$1F = 1 \text{ lot of } 16 \text{ plus } F \text{ lots of } 1$$

A hex digit represents 4 binary digits (0000 to 1111) and two hex digits can represent one byte. This simple segmenting of binary into hex is an improvement on octal representation as a means of viewing binary data in modern machines. However, it is worth bearing in mind that if computers still employed a 6-digit byte, rather than the ubiquitous 8-bit byte, then the octal system, where an octal digit is equivalent to 3-bits, would be better for segmenting and viewing binary numbers.



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

See the DEC PDP-10 which used a 36-bit word size.

<http://www.encyclopedia4u.com/p/pdp-10.html>

Example Usage

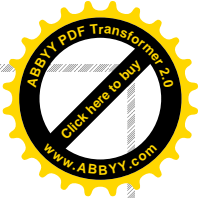
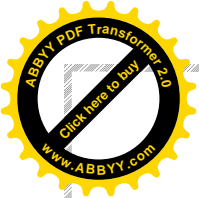
Hex has many applications, of which the following are just a few:

- RGB colour representation
<http://www.lynda.com/hexh.html>
- NIC (Network Interface Card) MAC (Media Access Control) addresses
<http://www.cityu.edu.hk/csc/deptweb/facilities/ctnet/wlan/machelp.htm>
- Extended ASCII Chart used for representing 28 or 256 characters (0 ..255), including all of those on your keyboard plus a few esoteric symbols. Any ASCII value can be represented by a 2 digit hex i.e. 00 .. FF. <http://www.asciitable.com/>

Conversion of Numbers

The following offers some rules for conversion of numbers between differing number systems. You may find other approaches that are more suitable (and understandable) e.g. the tabular layout outlined on the previous three pages.

Decimal	Binary	Hex	Octal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16



15	1111	F	17
----	------	---	----

Examples

1. Binary to Octal Conversion : Binary Number 1001001011110101

2. Split into groups of 3 binary digits (bits) starting from the right.

1 001 001 011 110 101 Binary
1 1 1 3 6 5 Octal

3. Binary to Hexadecimal Conversion : Binary Number 1001001011110101
Split into groups of 4 bits starting from the right.

1001 0010 1111 0101 Binary
9 2 F 5 Hexadecimal

4. Decimal to Binary:

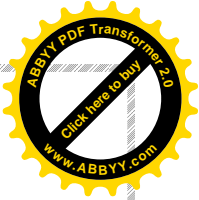
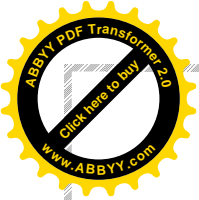
Example:
Convert 46(base 10) to binary.
Divide the number successively by 2 and note the remainders (0 or 1). Write down these remainders, starting from the right and working left.

46	/	2	rem	0	0
23	/	2	rem	1	10
11	/	2	rem	1	110
5	/	2	rem	1	1110
2	/	2	rem	0	01110
1	/	2	rem	1	101110
0					
Binary					0101110

Binary Addition

0	0	1	1
+0	+1	+0	+1
0	1	1	carry(1)0

If a carry is generated, it is carried one position to the left as in decimal arithmetic.



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

Example:

Add binary numbers 101 and 111

```
  101
+111
-----
 1100
```

Starting from the rightmost column, $1 + 1$ means that you write 0 and carry 1 to the next column on the left.

Move to the left, add $0 + 1$ plus the 1 you carried. Therefore you write 0 and carry 1 to the next column.

Move to the left, add $1 + 1$ plus the 1 that you carried. Therefore you write 1 and carry 1 to the next column.

Move to the left, nothing to add other than the 1 you carried. Therefore the 1 is placed in front of the other 3 binary digits.

Result 1100.

Number System Exercises

Left click on the A icon to get the answers (switch off pop-up blocker if you have one enabled).

1. Convert the following IP addresses into 32-bit binary form.

(i) 127.0.0.1

(ii) 199.34.57.255



2. Convert the following decimal RGB values into hex form.

(i)

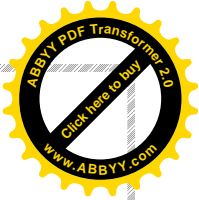
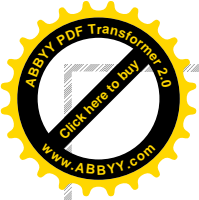
R 255

G 000

B 051

(ii)

R 102



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

G 204

B 153

A

3. Add the following binary numbers

(i) $00010110 + 00111111$

(ii) $01111101 + 01100011$

A

4. Convert the answers to 3(i) and (ii) into decimal numbers.

A

5. By using the 8-bit ASCII table, the following binary code can be used to represent the four characters M150

01001101 00110001 00110101 00110000

Give the decimal representation of each character.

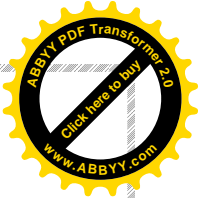
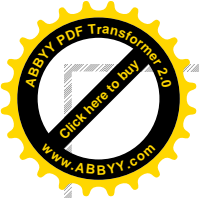
A

An Introduction to Document Markup

Early attempts at the transformation of paper documents to electronic documents were plagued by problems.

- Employed procedural markup - Mark this text as 12 point Times New Roman, then mark this up as bold etc.
- Focus was on presentation rather than document structure

Introduction of macros, to produce generic coding and speed up processing



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

- A set of commands/keyboard strokes that can be represented by a single name or tag. (e.g. format a heading in some specific way)
- Automated process enables swifter editing - a single change to the macro can update all related parts of a document structure (e.g. headings need to be reformatted)
- Macros represent an authoring shorthand

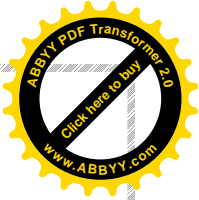
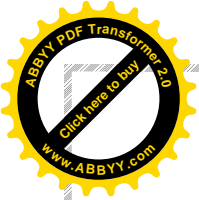
However, formatting was proprietary (still is in many cases e.g. MS Word versus Adobe Acrobat) and this made the transfer of documents, even within the same company/organization, at best difficult and often impossible.

IBM's Goldfarb, Mosher and Lorie (GML), identified the need for:

- A common document format
- The requirement that any format must be specific to a domain (in GML's case, legal documents were their concern)
- Reliability; to achieve this formatting of documents must adhere to a set of rules

GML invented Generalized Markup Language (coincidence?).

- Based on the then current form of document markup (tags consisting of names and symbols). The semantics (meaning) of the tags in any GML compliant document would be universally understood.
- It would provide a language specific to the users' needs. For example, legal documents have a language of their own. Context is important and this is sometimes referred to as the "Domain of Discourse". This, together with a specified language, provides the means for successful exchange of information between any interested party. The "Domain of Discourse" should remove any potential for ambiguity.
- For successful computer processing, there is a need for markup to be verified according to a pre-determined set of rules, or constraints.



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

Standard Generalized Markup Language (SGML)

Goldfarb developed GML over a number of years. Eventually GML became standardized and recognized as an international standard in 1986. Known as Standard Generalized Markup Language, SGML.

SGML provides a framework for developing document structure and conformance. SGML has:

- No specific document structure
- No limits on the tag set
- No constraints on the ability of users to develop new document standards

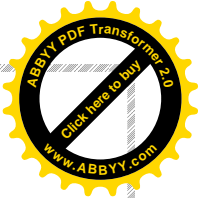
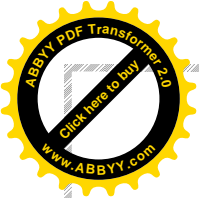
SGML found immediate use in military documentation and in the civil aviation industry.

The advantages of structured data, without regard to document presentation:

- Data can quickly accessed and selected according to some pre-determined criteria
- Data can be readily sorted
- Data can be transformed
- Required data can be output to a variety of media (print, screen - PC, Mac, PDA, mobile phone - audio)

HTML and Hyperlinking

[Tim Berners-Lee](#) understood the need for electronic document delivery at CERN. He had witnessed the problems encountered by physicists attempting to find information relevant to their work ("the technical details of past projects are lost forever"). Duplication of projects, the inability to access data due to differing terminals and software, and differing document



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

formats (SGML, Unix, CERNDoc etc) were some of the most obvious problems.

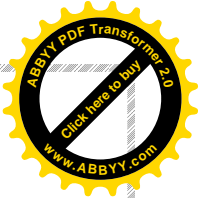
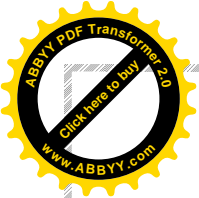
Already familiar with SGML, he produced a simple subset for formatting documents and named it Hypertext Markup Language, HTML. In 1990 he and Robert Cailliau wrote software for viewing these documents. This browser was named the WorldWideWeb (one word). Later renamed as Nexus to differentiate it from the "information space", which was now named the World Wide Web.

Berners-Lee promoted the concept of "universal readership":

- One program developed to access data
- Can be read by anyone, anywhere and on any machine

The sequential access of information was considered to be inadequate; linked information was the way forward.

- Concept of linking documents already understood by others ([Bush](#), [Englebart](#), [Nelson](#)). TBL's idea was to **hyperlink** CERN's electronic documents.
- DTD was not written for his first version HTML 1 (TBL wanted to keep it simple; there would be no need for knowledge of SGML)
- HTML tags do not convey semantics of content, just document structure
- Simple tag set e.g. `<h1></h1>` for main heading; `<p></p>` for a paragraph.
- A DTD was written for HTML 2. Didn't last long before it was "hijacked"
- Mosaic browser written. First attempt at customizing HTML. Included an image tag to display images in Web documents. Tim is not pleased.
- Browser Wars! Netscape (Mosaic) versus MS Internet Explorer. Each new version offered additional features, hence moving further away from TBL's HTML.



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

- A return to the same old problems; namely proprietary software and a lack of common formats.
- HTML becomes "bloated" and fails to provide document structure by concentrating on document appearance.

Berners-Lee and others (W3C) attempt to stop the rot by returning to standards

Introduction to eXtensible Markup Language (XML)

XML is:

- A subset of SGML
- Developed for deployment over the Web (SGML is too complex for Web apps)
- Not a replacement for HTML
- As easy to use as HTML, but with the benefits of SGML. In other words it can be easily written in a text editor and yet it is formal and concise.
- Multi-platform (can be deployed on PC, Mac, PDA, mobile phone etc)
- Easily read - a text file
- Extensible - users can develop their own tags to convey the semantics of content.
- Non-proprietary

Here is an example of how data can be structured using XML.

```
<? Xml version="1.0" encoding="ISO8859-1"?>
```

```
<CATALOGUE>
```

```
  <BOOK>
```

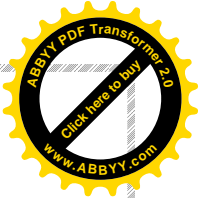
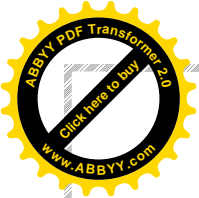
```
    <TITLE>Mystic River</TITLE>
```

```
    <AUTHOR>Dennis Lehane</AUTHOR>
```

```
    <GENRE>Crime Fiction</GENRE>
```

```
    <PUBLISHER>Bantam Books</PUBLISHER>
```

```
    <YEAR>2001</YEAR>
```



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com
mobile :0551783558

</BOOK>
</CATALOGUE>

This XML file will hold data relating to a book collection. Note that an XML document must have a root element, in this case `CATALOGUE` (I could have called it anything I like - that's why it's eXtensible Markup Language). The nested element, `BOOK`, is a child element of `CATALOGUE`. Other child elements (`TITLE`, `AUTHOR`, `GENRE`, `PUBLISHER`, `YEAR`) have been further nested within the `BOOK` element. For simplicity I have included just one `BOOK` element. As with a relational database, this file could store thousands of identical data structures, each one representing the required data for a book.

Transforming XML data with XSL

Data that has been structured with XML can be transformed by applying programming instructions using XSL (eXtensible Stylesheet Language). XSL, a family of languages (XSLT, XPath, XSL FO), is itself written using XML syntax. An XML document has a tree-like structure for its data. This is referred to as a **source tree**. XSL transforms the source tree into another tree, a result tree, before applying formatting instructions for output. I have avoided discussing the use of templates and pattern matching, which XSL uses to implement the creation of a new result tree.

I have created some files for you to use. Remember to store them in the same folder.

You can get an XML file. (Right click and Save Target As...). This is simply an extension of the markup that you saw on the previous page. It stores the data for a number of books.

You can get an XSL file. This is the program that will be used to transform the XML data.

Reading these files is easy. Open Notepad, File/Open and select Files of type: All Files before navigating to the `book.xml` file. Repeat the process for the file `book.xsl`.

The XSL transformations are a series of instructions that operate on the XML data (source tree). By accessing particular elements of the source tree, a

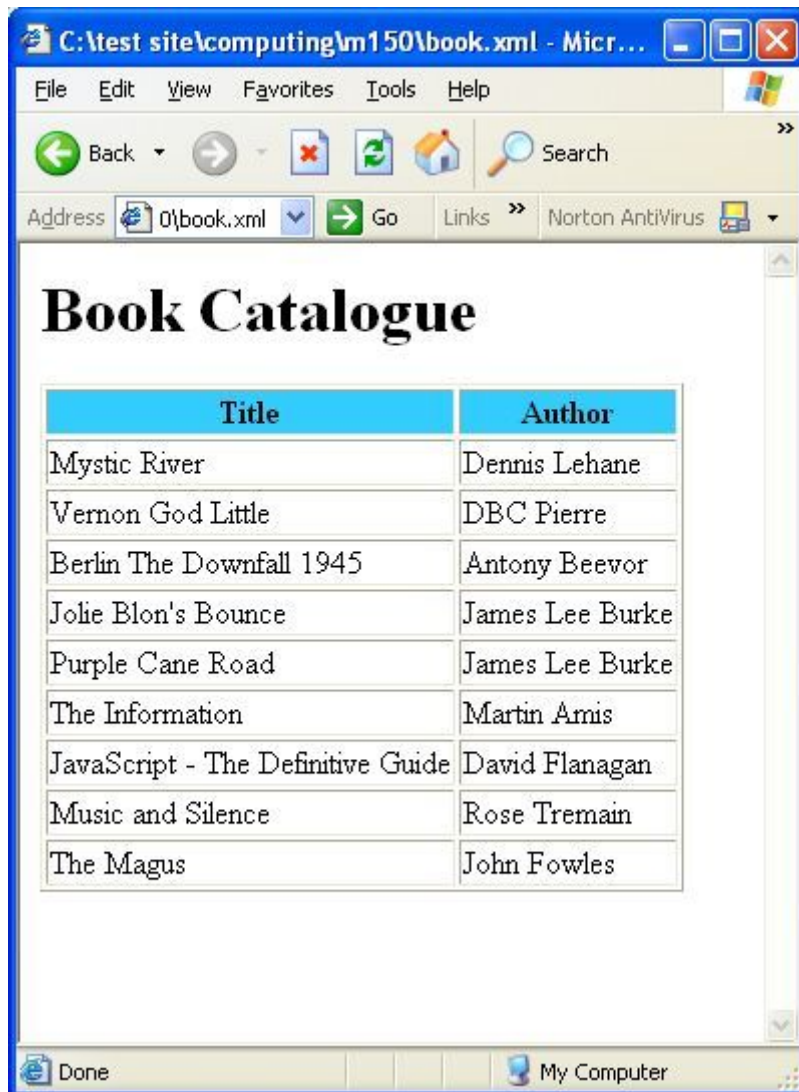
Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

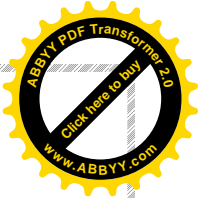
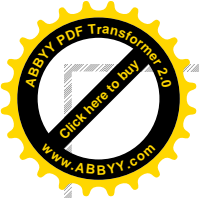
mobile :0551783558

new tree is created that uses these elements in some predetermined manner. Once that new result tree is constructed, the elements can be displayed in a browser with HTML. A demonstration of this will provide another example of the transformation of data into information by use of a programming language.

Using IE6, which supports XSL, open `book.xml` (File/Open and Browse to select the file). As long as both files were saved in the same folder, you should see that the XSL file has transformed the XML data into this:



The XSL instruction/element `<xsl:for-each select="CATALOGUE/BOOK">` contains a `for-each` instruction, which represents a looping construct common to many programming languages. The `select="CATALOGUE/BOOK"` part of the instruction enables us to



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

access and operate on each of the nested `BOOK` elements. This harks back to the point about structuring data or documents. The predictability of a data structure, in this case an XML data tree, enables us to write programs for operating on that data.

Once each `BOOK` element is accessed, each of the values of the four nested elements can be accessed in turn. However, the XSL file instructs the browser to select only two values, that of `TITLE` and `AUTHOR`, with the next two lines

```
<td><xsl:value-of select="TITLE"/></td>
```

```
<td><xsl:value-of select="AUTHOR"/></td>
```

Note that the `<td></td>` tags are used to construct the cells of a HTML [table](#). The XSL is simply getting the XML data and placing it in HTML tags for output to the browser.

Filtering XML Data

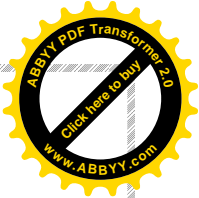
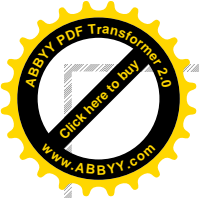
We can choose to view data by applying a filter. Download and save this [file](#), which contains some modifications to the XSL that you saw on the previous page. To link this XSL file to the XML data, a small change is required to the `book.xml` file. Open `book.xml` in Notepad and change the line that reads

```
<?xml-stylesheet type="text/xsl" href="book.xsl"?>
```

to read

```
<?xml-stylesheet type="text/xsl" href="book2.xsl"?>
```

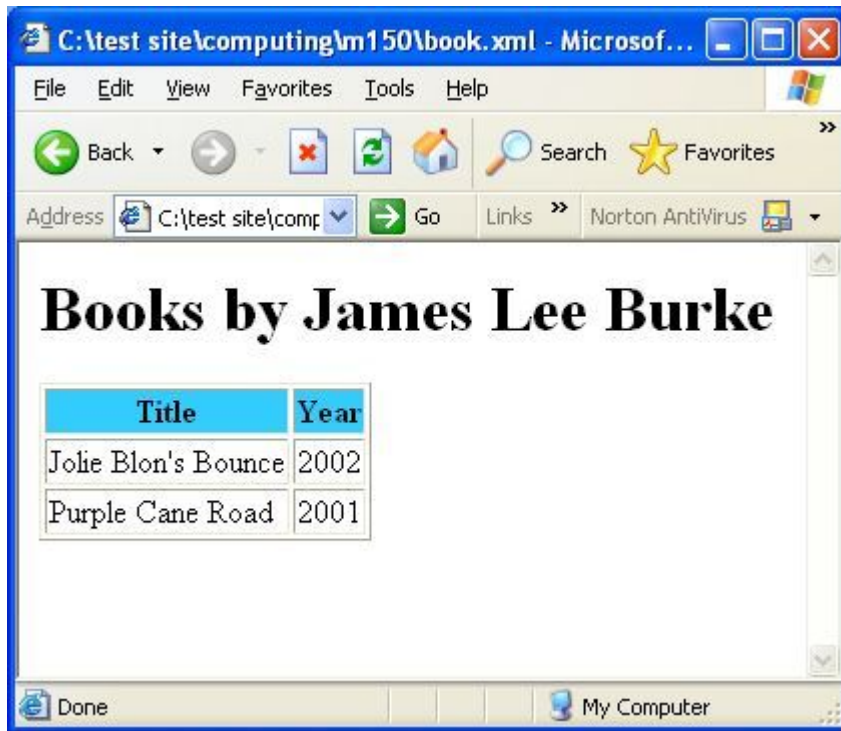
Save the change and then open `book.xml` in your IE6 browser.



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558



Look at the XSL instruction (open book2.xsl in Notepad)

```
<xsl:for-each select="CATALOGUE/BOOK[AUTHOR='James Lee Burke']">
```

You have already seen the `for-each` and `select` instructions. The filter `AUTHOR='James Lee Burke'` is used to access data only where the corresponding nested value of the `AUTHOR` element is `James Lee Burke`; anything else will be ignored. Once this is achieved, the next two lines get the values for `TITLE` and `YEAR` and insert them into the cells of a table

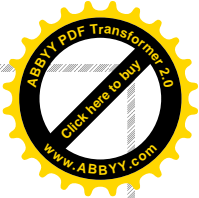
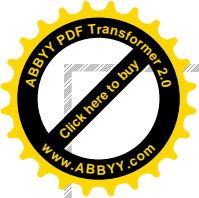
```
<td><xsl:value-of select="TITLE"/></td>
<td><xsl:value-of select="YEAR"/></td>
```

Sorting XML Data

Now try sorting the data by applying an instruction to sort by value of the `TITLE` element. Download and save this [file](#) and modify your `book.xml` file with this line

```
<?xml-stylesheet type="text/xsl" href="book3.xsl"?>
```

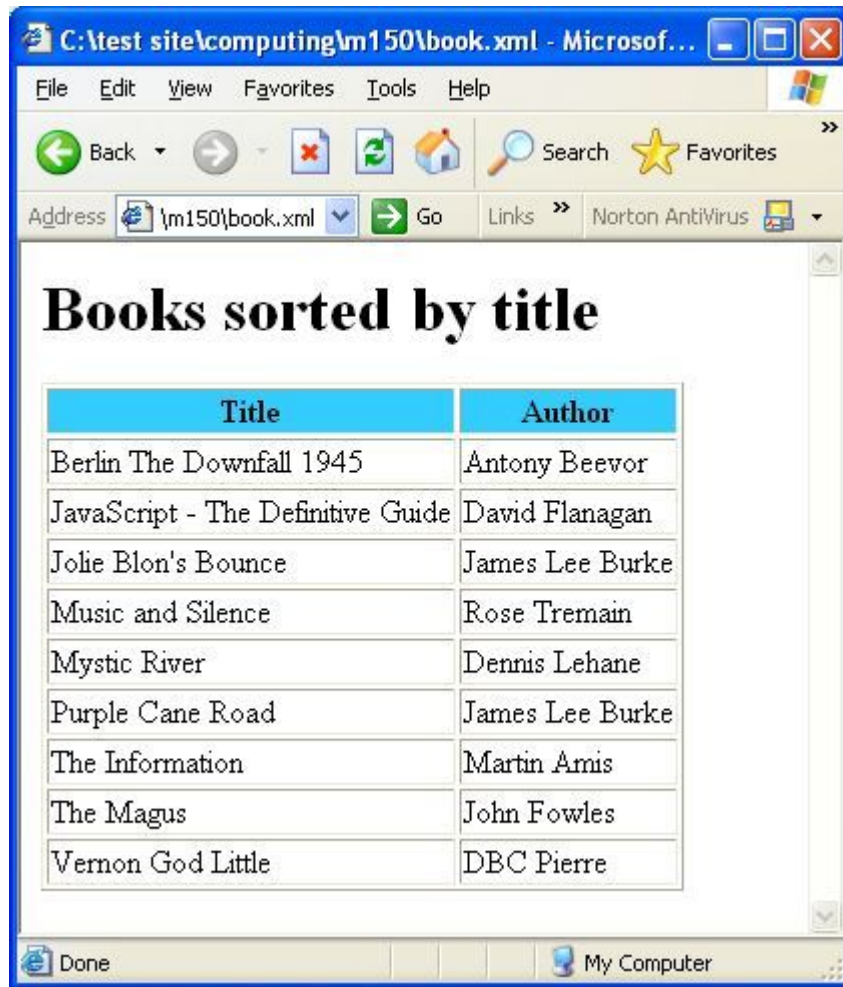
Reload `book.xml`



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558



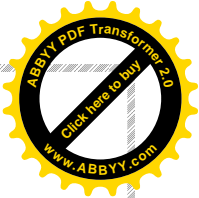
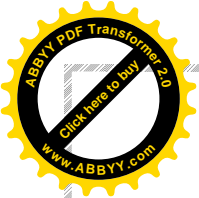
The relevant XSL instruction is:

```
<xsl:sort select="TITLE"/>
```

Try sorting by the name of the author and remember that where XML is concerned, lowercase `author` is not the same as uppercase `AUTHOR`. The correct syntax must be adhered to, or your code will fail. THE XML elements were written in uppercase, although, because XML is extensible, I could have used lowercase. Note that this is not to be confused with writing XHTML tags, which must be written lowercase.

Applying selection to XML data

Data can be selected according to some criteria or condition. This is a common programming mechanism and you will use it in JavaScript programming.



Prepared by Mr.Emad Al-Shafey

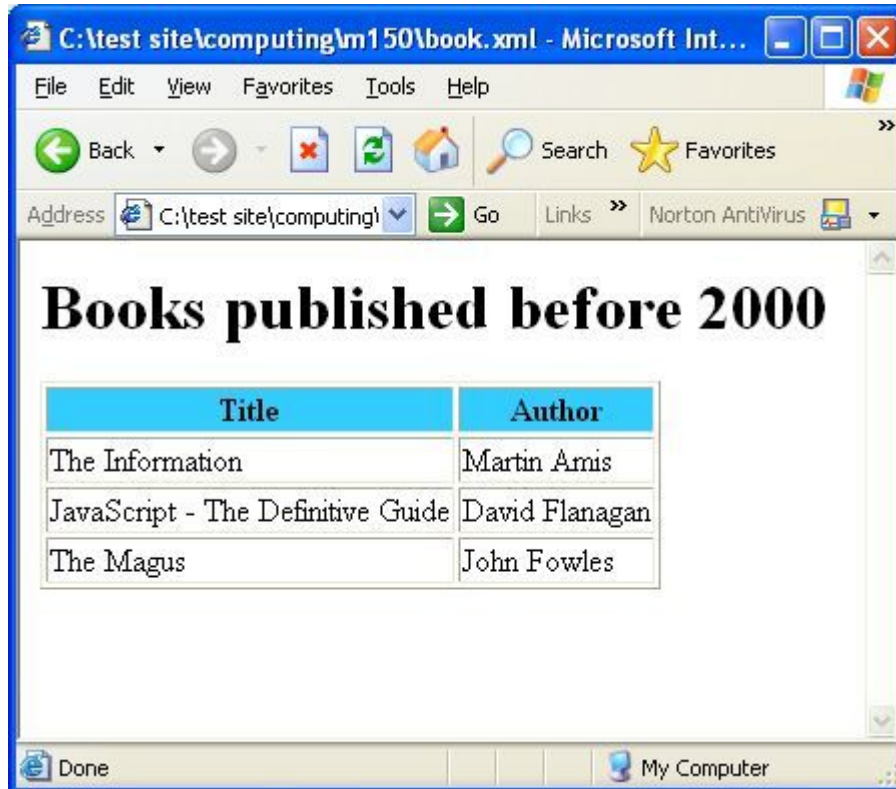
a_one31@hotmail.com

mobile :0551783558

Download and save this [file](#) and modify your `book.xml` file with this line

```
<?xml-stylesheet type="text/xsl" href="book4.xsl"?>
```

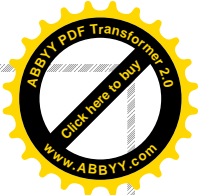
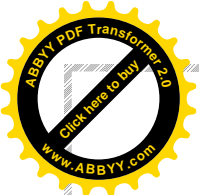
Reload `book.xml`



In this case if the value of the `YEAR` element is less than 2000, then the values of the `TITLE` and `AUTHOR` elements will be selected for output. Note that:

- The XSL syntax for "is less than" is `<` in the line
`<xsl:if test="YEAR < 2000">`
 You can try other selections using
`=` meaning equal
`!=` meaning not equal
`>` meaning is greater than

- Unlike the instruction for sorting, the XSL `<xsl:if test>` element requires a matching `</xsl:if>`



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558

Applying more complex selection to XML data

In common with other programming languages, XSL provides you with the language/syntax to write more complex conditions and uses the keywords `choose`, `when`, `otherwise`.

Note: JavaScript will use a different syntax, but the effect is the same

```
if (some condition)
```

```
{
```

```
    doSomething
```

```
}
```

```
else
```

```
{
```

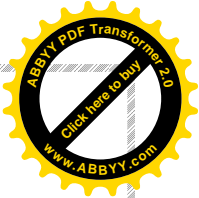
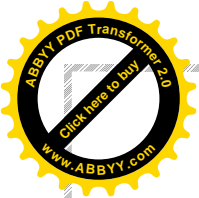
```
    doSomethingElse
```

```
}
```

Download and save this [file](#) and modify your `book.xml` file with this line

```
<?xml-stylesheet type="text/xsl" href="book5.xsl"?>
```

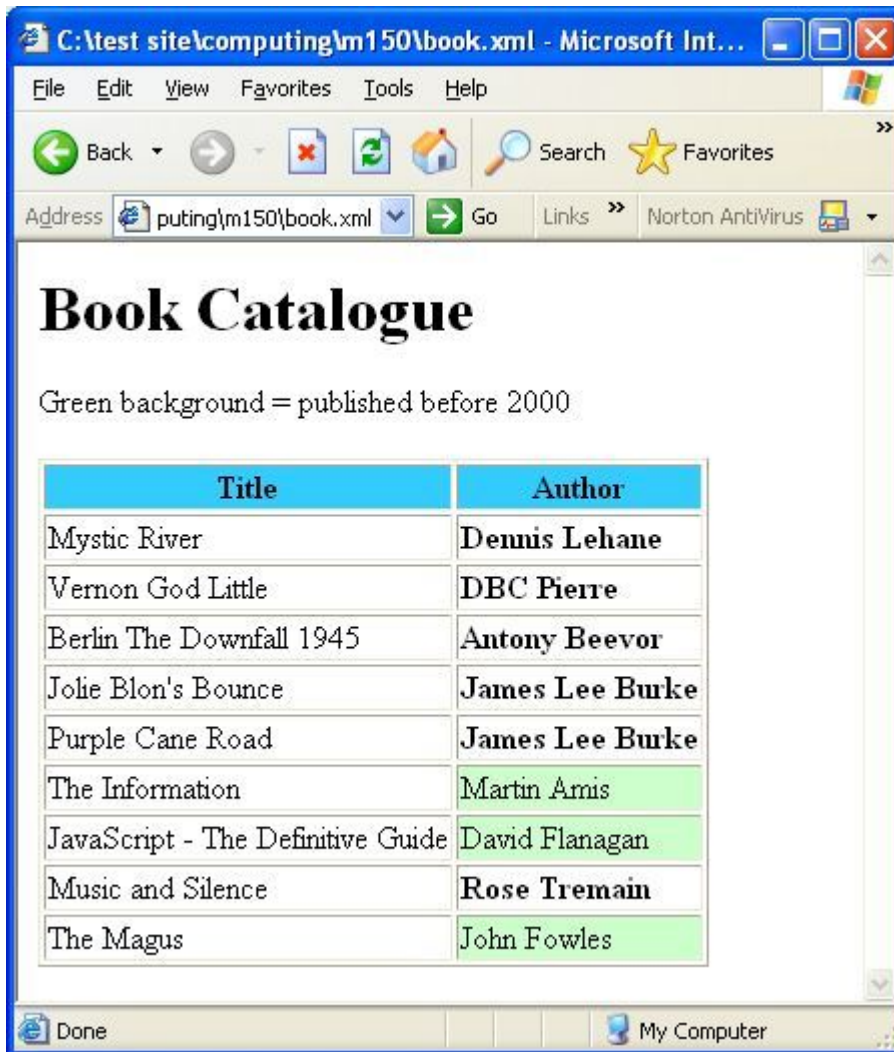
Reload `book.xml`



Prepared by Mr.Emad Al-Shafey

a_one31@hotmail.com

mobile :0551783558



This tests for a `YEAR` element greater than 1999. If it is found, the corresponding `AUTHOR` value is output in bold font (the `` tags are used for this). If the `YEAR` value does not meet this condition, the background colour of the table cell is coloured pale green, `<td bgcolor="#CCFFFF">`.