The Open University

TM111 Introduction to computing and information technology 1

Block 1

The digital world

This publication forms part of the Open University module TM111 *Introduction to computing and information technology 1*. Details of this and other Open University modules can be obtained from Student Recruitment, The Open University, PO Box 197, Milton Keynes MK7 6BJ, United Kingdom (tel. +44 (0)300 303 5303; email general-enquiries@open.ac.uk).

Alternatively, you may visit the Open University website at www.open.ac.uk where you can learn more about the wide range of modules and packs offered at all levels by The Open University.

The Open University, Walton Hall, Milton Keynes MK7 6AA

First published 2017.

Edited and designed by The Open University.

Typeset by The Open University.

Printed in the United Kingdom by Hobbs the Printers Limited, Brunel Road, Totton, Hampshire SO40 3WX

ISBN 978 1 4730 2102 0

1.1

Contents

Part 1 Living in a digital world

by Elaine Thomas

Contents

Introduction

Welcome to the first block of TM111 *Introduction to computing and information technology 1*.

As you study Block 1 *The digital world*, you'll experience and investigate a wide range of aspects of the digital world. Many of the things we enjoy – such as music, images, email and social networking – are based on digital technologies, and you will see how advances in these technologies have led to them playing an increasingly significant role in our lives. You will examine some practical examples, and explore some of the issues and debates that surround the use of digital technologies in a range of everyday situations. Block 1 also introduces some of the key skills for study, including communication, numeracy, practical and professional skills, and digital and information literacy (DIL) skills, to help you become an effective learner.

Part 1 *Living in a digital world* explores what is meant by the 'digital world' in terms of the technologies involved, and introduces some of the legal, social and ethical considerations. Part 2 *The evolving computer* discusses how computers have developed since the early days to the present time when computers are 'ubiquitous' in that they are everywhere around us. The idea of using binary numbers to code digital representations is also introduced. Part 3 *Digital media* looks at how images and sound are represented as binary numbers in computers and you will use a sound editing program, called Audacity, to record and manipulate sound. Part 4 *A world built of data* takes the idea of data representation further, exploring the relationship between data, computers and humans. Part 5 *Weaving the web* explains how the web works and includes hands-on work building simple web pages using HTML codes and editing tools. Part 6 *Crossing boundaries* looks at human computer interfaces (HCI) as the shared boundaries between the user and the computer, and introduces you to concepts such as 'usability' and 'accessibility'. You will start your personal and professional development planning (PDP) at the end of Part 6. You will then complete and submit your tutor-marked assignment (TMA).

Some of the topics you study over the next few months may be familiar to you, but others will be new. I hope that you find most, if not all, of them interesting and enjoyable.

The module website

The module website includes an online study planner which will guide you through the printed and online study resources and help you to plan your work. You should visit the module website *before* continuing with Part 1 and *before* starting each subsequent part.

Some activities require you to be online and this will be indicated by an icon (as shown) in the margin. Words in **bold** are included in the glossary, which you will find on the module website. If you haven't already done so, make sure you look at the TM111 *Module guide*.

Introducing Part 1

My job in this first part of TM111 is to start you off on the digital journey that you'll be making over the next few months. You will learn a little about the historical development of communication technologies and computers and their role in today's society, and you will consider examples of digital technologies in the world around you and their influence on your life. However, this part merely introduces some of the key ideas; you will study them, and others not mentioned here, in greater depth later on.

At this stage you are probably wondering what we mean by the title of Block 1 *The digital world*. To begin, you'll need to know that a digital technology is any technology based on representing data as sequences of numbers. This is a fundamental idea and you will learn much more about it as you work through TM111. In this module, the term *digital world* refers to the influence of computers on the world, in a wide variety of forms, linked by communication networks.

Most of the uses of technology that you'll consider will be legal and harmless. However, you will also look at some examples that subvert laws and the rights of others. Other examples move into the worlds of crime and terror where information is transmitted that others want to intercept, control or suppress. Effective participation in the digital world therefore requires you to become aware of many technical, social and ethical pitfalls, as well as some of the ways you can avoid them.

Activity 1.1 (exploratory)

How digital is your world? You should now complete the 'How digital is your life?' quiz which you will find on the module website. Note: there are no right or wrong answers in this quiz.

Comment

As I said above, there are no right or wrong answers to this quiz, so some of the questions may not seem relevant to you. As you study TM111 you will learn about new ideas, develop new skills and become more confident. You might like to return to the quiz towards the end of TM111 to see whether your responses to the questions have changed.

Computers and communications: from rarity to ubiquity

1.1

The theme of this section is the development of **digital technologies**, from the physically large and highly expensive equipment of the 1950s to the omnipresent **networks** and computer-based devices upon which our digital lives are founded today. In this section I will:

- compare the development of the computer to the development of the telephone
- describe some of the digital technologies that form an integral part of many of our lives
- introduce the term *information society*.

1.1.1 The telephone

In the 1950s, telephones represented the cutting edge of technology. However, they were not only large and expensive to purchase but also inconvenient to use; if you wanted to make a call over a distance of more than fifteen miles, you had first to call the operator who would then make the connection.

Telephones were also expensive to use. In the UK, calls were charged in units of three minutes, each unit costing the equivalent of between 2 and 19 pence in decimalised currency. (In comparison, in 1955 a pint of milk was cheaper at about 3 pence.) Few people had a home telephone and even fewer made long-distance calls so most people made use of public telephone boxes (Figure 1.1). The high cost of using a telephone ensured that anyone who had one tended to use it infrequently, and then only for short calls.

Figure 1.1 Inside a public telephone box

In 1959, the General Post Office, which ran almost all UK telephone systems at the time, introduced subscriber trunk dialling (STD). This advance in technology allowed users to make long-distance calls directly, without an operator, and to be charged only for the actual duration of the call. The telephone became easier and much cheaper to use; as a result, more people began to use it and for longer calls.

The first ever public call on a mobile phone was made in 1973. The call was made on a Motorola device resembling a brick at 22 cm long, weighing about 1 kg and with a talk time of just over 30 minutes.

Over time, the telephone has become part of the background of our lives. Nowadays it is extremely unusual to find someone who has neither a home telephone nor a mobile phone. Just as the telephone changed from a status symbol to become simply another piece of the modern world, so the computer has made a similar transition.

1.1.2 The computer

There were very few computers in the 1950s, and those in existence were treated as objects of wonder with almost mythical powers. They were nothing like the computers of today, as you'll see in a later part. For one thing, they were huge, such as that shown in Figure 1.2 with Grace Hopper, inventor of the computer language COBOL (COmmon Business-Oriented Language). They were also delicate and consumed a lot of electricity, wasting much of it as heat.

Figure 1.2 Grace Hopper beside a 1950s mainframe computer

Nowadays, however, a computer is just another item stocked in supermarkets. And as computers have become cheaper and smaller, they have been incorporated into a kaleidoscopic range of devices that bear no resemblance to what was once thought of as a computer. Powerful computers now sit at the heart of objects as diverse as smartphones and games consoles, cars and vacuum cleaners. The cost of computer power continues to decrease, making it possible to incorporate computer technologies into almost any object, no matter how small, cheap or disposable. And these **smart devices** are ‘talking’ to one another – not just within a single room or building, but across the world via the **internet**, using the **world wide web** (see Box 1.1). Thus even as the computer vanishes from sight, it becomes vastly more powerful and ever-present – to use a term you’ll become very familiar with, it is now **ubiquitous**.

In the smartphone, two previous separate technologies, computers and the mobile telephone systems, have become combined in one device. This **convergence** of computing and telephone technologies has resulted in new goods and services for people to buy and use. For example, many people who were not previously interested in computers are happy to take photographs on their mobile phones and share them. In this situation, the computer in the phone handles the processing of the photograph and enables it to be sent by email or uploaded to social media.

Activity 1.2 (exploratory)

Can you think of another technology that has made the transition from novelty to commonplace, like telephones and computers?

Comment

There are many possible answers to this question. I thought of televisions; there used to be only one per household, but today homes often have more than one television – and television screens are available in public spaces such as health centres, hotels and even in parks. Also, people can watch television on their computer or a tablet through a broadband connection.

Box 1.1 The internet and the web

You will come across the terms *internet* and *web* a lot in TM111. Although in everyday life people tend to use these terms interchangeably, in reality they are two separate (though related) entities.

The internet is a global network of networks: an *internetwork* (hence its name). It is the infrastructure that connects computers together. At first written with an upper-case ‘I’, it is now usually seen with a lower-case ‘i’.

The web (short for *world wide web*), on the other hand, is a service that links files across computers, allowing us to access and share information. Thus the web is a software system that has been built upon the **hardware** of the internet.

Apart from anything else, this means that it is technically incorrect to refer to ‘searching’ or ‘browsing’ the internet. When you carry out an online search, you are in fact searching the web!

1.1.3 The networked world

Networks and the internet

As a result of advances in **information and communication technology (ICT)**, our notions of time and location are changing – distance is no longer a barrier to commercial or social contact for those of us connected to suitable networks. Some people may find it difficult to imagine not having access to the information and services that play a crucial part in their daily lives. Others may feel that they have no part to play in the digital world because their network access is very limited or even non-existent. Some simply don't care about the digital world, viewing it perhaps as a waste of time. Yet whether we are aware of it or not, digital information is flowing constantly around us.

Consider a computer that is connected to the internet – the one you intend to use in studying TM111, for example. This may be a computer you use at home, in a library or at work; you may use it on the move or in a fixed location. Whatever the case, this computer is part of a complex system consisting of wires and optical fibres, microwaves and lasers, switches and satellites, that encompasses almost every part of the world. The oceans are wrapped in more than a quarter of a million miles of fibre-optic **network cable** with several strands of glass running through it. Each of these strands can carry thousands of simultaneous telephone conversations, a few dozen television channels, or any of a range of other forms of digital content (such as web pages).

This modern communications network enables us to use a mobile phone in the depths of Siberia or take a satellite telephone to the Antarctic (Figure 1.3), watch television in the middle of the Atlantic, do our banking on the train, or play games with a person on the other side of the world. It is one of the greatest technological achievements of the last thirty years and it is so reliable, so omnipresent, that we very rarely stop to think about what actually happens when we dial a telephone number, click on a web link or switch TV channels. Or, rather, we tend not to think about it until something disrupts the network – whether it be a widespread problem such as a power cut, or something more localised such as finding ourselves in a rural area with slow broadband access or no mobile phone signal.

Figure 1.3 Using a satellite phone in the Antarctic

Information, knowledge and learning

The end of the twentieth century and the beginning of the twenty-first century are often compared to other historical periods of great change, such as the Industrial Revolution, because of the huge technological changes that are happening in many areas of our lives. These developments are taking place in conjunction with correspondingly large social and economic changes, often characterised by the terms **information society** and **network society**. Policy makers frequently refer to such terms when driving forward changes in the technological infrastructure. Politicians often refer to the inevitability of technological change in our information society and stress the need to be at the forefront of these changes in order to secure future prosperity. The development of the broadband network infrastructure, making public services available online, and equipping schools and local communities with computers are examples of such changes.

Another term you might have encountered is that of the **knowledge society**, which refers to the way that new information systems can transform human societies. Included in the knowledge society is the idea of the 'learning society'. The pace of change is so rapid nowadays that learning can no longer be confined to our school years and early adulthood. Everyone must continue to learn throughout their adult lives in order to benefit from the economic opportunities that rapid development makes possible.

Activity 1.3 (exploratory)

Can you think of an example where changes in technology have resulted in changes to your work, social or family life? Have those changes improved your life? Have they created any problems?

Comment

The biggest change for me has been in my ability to work from home. I'm currently sitting at home typing this text on a computer whilst listening to some music, which is also stored on the computer. I'll shortly email my document to a colleague, who will be able to read it a few seconds later.

I can also share my document with colleagues through an online storage system.

Being able to work from home has its advantages and disadvantages. I find it easier to concentrate when I'm working at home, rather than in a busy office space shared with a dozen other people. It also saves me the time and money that used to be spent on travelling. On the down side, the boundaries between my home life and my work life have become very blurred, as my family will confirm.

The rise of texting and messaging

You might wonder about how these technologies change the way we live, work and relax, but sometimes it works the other way around. Unintended uses sometimes develop alongside the intended uses of emerging technologies in our information society. A classic example is the text-messaging facility on mobile phones, often referred to as **SMS** (which stands for short message service). This was originally a minor feature designed to be used by engineers testing equipment – it was not expected to be used by phone owners at all. Yet in the first decade of the twenty-first century, SMS messages were one of the most profitable parts of the mobile phone business (International Telecommunication Union, 2006). SMS resulted in a whole new method of communication and form of popular culture, different ways of interacting with radio and television, and even a new language form: texting.

Income from SMS messages peaked at around US$104 billion in 2011 (Associated Press, 2012). In more recent times, **instant messaging (IM)** 'apps' used on smartphones, such as WhatsApp, Snapchat and Telegram, have overtaken SMS in popularity. By 2015, the IM company WhatsApp was handling 30 billion messages every day, compared to the 20 billion SMS messages sent daily (Evans, 2015). This difference in usage is partly because there is no extra cost involved in using IM services. SMS messages can be expensive on some contracts and require a mobile phone signal, but they are still in use. Multimedia messaging services (MMS) allow for text, audio and images to be sent via the mobile phone signal. The total number of SMS and MMS messages sent in the first three months of 2016 was 24.1 billion, a decrease of 1.8 billion messages (6.8%) compared to a year previously (Ofcom, 2016). These services and applications are blurring the boundaries between texting and online communication.

Note: 'billion' is a word that in the past had different meanings in the UK and the USA. However, the two countries now agree that a billion is one thousand million (1000 000 000), and TM111 will follow that convention.

Study note 1.1 Interpreting charts

A bar chart is a way of representing numbers pictorially and can illustrate patterns or trends in the data. In a bar chart, the numerical values are represented by the height or length of the rectangles. Their particular strength is enabling us to make comparisons. The chart in Figure 1.4 represents the percentage of internet users who used instant messaging in 2005 and in 2014 according to age group.

All charts and graphs follow a set of conventions. The charts should include a title explaining what is shown or being represented. The two values being described in Figure 1.4 are the percentage of internet users and the age group. Each axis should be labelled clearly to indicate what units are being used, and have a marked scale. In this example, the vertical axis represents percentage of internet users on a scale of 0% to 90% and the horizontal axis shows the age group.

As with all types of graph, it does not offer any explanations about why the data is how it is. You will have noticed that use varies by age group and that there has been a considerable increase in use over time across all age groups.

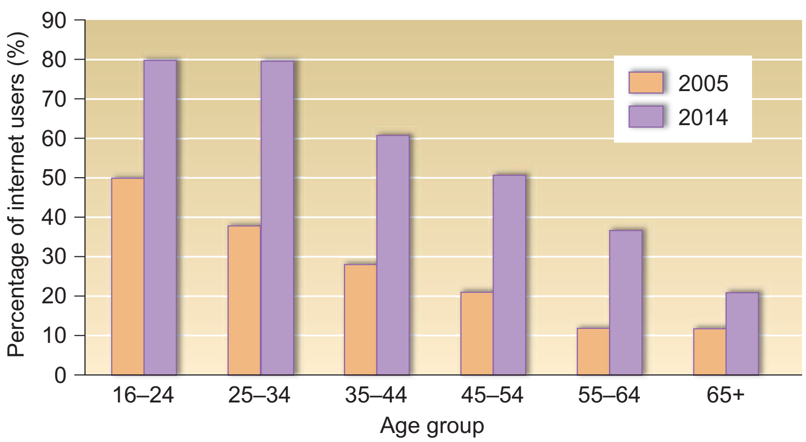


Figure 1.4 Weekly use of the internet for instant messaging, by age group: 2005 versus 2014 (Source: Ofcom, 2015)

Activity 1.4 (self-assessment)

(a) Study the chart in Figure 1.4. Which age group shows the smallest increase in the percentage of internet users who instant messaged between 2005 and 2014?

(b) Which age group shows the biggest increase in the percentage of users between 2005 and 2014?

(c) Can you offer any explanation for the increase in use amongst all the age groups?

Note: as in all the printed parts of TM111, answers to self-assessment activities are provided at the end of the part.

Study note 1.2 Maths skills

If your maths skills are a bit rusty, then you may wish to refer to the *Using numbers* booklet which provides information on various mathematical concepts, including percentages.

Study note 1.3 Making notes as you study

When you have finished reading a section of a part, it can be useful to take time to check whether you've understood the main points. This is also an opportunity to go back over some of the relevant sections and make some notes. The purpose of making notes is twofold: to check that you've identified the main points from a section and to check your own understanding. If you just read passively, you risk losing concentration. Extracting the main points from a piece of writing and then writing them as notes in your own words is the best way of making sure that you have understood.

An effective method for making notes is to start by skim-reading (a quick glance through to get an overview and some idea of the main points) and then re-reading the material more thoroughly, underlining or highlighting the main points as you go along. Using these underlined or highlighted notes, you can then summarise the main points by making a list using some of your own words.

Notes can take different forms; some people like to make linear notes with bullet points, and some others create spray diagrams. The choice is yours. There are also note-making apps that you can use when you are studying online material. You should find that developing your note-taking skills will help you to understand and remember key ideas. You will find further advice on note-making and the different types of notes you can use on the module website.

1.1.4 Conclusion

In this section, having briefly considered the rapid development of computers, I outlined some of the ways in which digital technologies pervade the world around us, giving rise to the social and economic changes that characterise our information society. The concept of the digital world is a key one in TM111, and in the next section I'll try to give you a better idea of what it involves.

Some aspects of our digital world

1.2

In the previous section, I introduced the concept of a digital world; in this section, I will outline some key aspects of such a world. I will look at how it affects business, information, entertainment, public services and communication. Many of the ideas you meet here will reappear several times during your study of TM111.

One of the things you may notice while studying TM111 is that some of the examples you read about no longer exist, or are nothing like as important as when the material was written. This is inevitable, and you shouldn't be concerned by it; your focus should be on the general principles, which remain valid. As an example, some companies you read about may cease to exist, and some may be more or less prominent. Yet the likelihood is that although companies come and go, there will still be ways to share photos (as Flickr and Google Photos allow at the moment), to edit documents online (as is possible with Microsoft Office 365 and Google Docs), to store files online (with file-hosting services such as Dropbox), and so on.

1.2.1 Business

Financial services

Every time you use a debit or credit card in a shop, the shop till communicates with a card terminal that transmits your identification details from your card to your bank or credit card company for verification. Your balance is then adjusted according to your purchase. A similar chain of events is initiated if you shop online (buying a plane or train ticket, perhaps) or over the phone (when buying a cinema ticket, for example). Many banks also provide online banking services, reducing the need for customers to visit a branch. **Automated teller machines (ATMs)** allow you to check your bank balance and withdraw cash wherever you are in the world. In each of the above situations – using a debit or credit card, shopping or managing your money online or over the phone, or using an ATM – the machines involved are connected via a network to a central computer which has records of your account in an electronic filing system known as a **database**.

Financial services have undergone huge changes in recent years as a result of developments in the digital technologies driving them. The examples just described show how convenient and accessible such services have become. Yet at the same time, issues of identity and security have become a concern. New online ways of communicating have also created new types of crime, including identity theft and digital financial fraud. In turn, these problems have fostered the development of new security industries that try to inform us and sell us solutions to reduce the chances of us becoming victims of online crime. Also, this move towards online services means that banks have closed many branches, leading to job losses. This is particularly difficult for customers who prefer to use branches, especially those living in rural areas.

Commerce

Advances in digital technologies have led to changes in many areas of commerce, with some existing kinds of business being transformed by the opportunities these developments offer. One of the most obvious changes is the emergence of retailers such as Amazon, which have an online shop but no physical one that customers can visit. Yet the internet not only benefits the largest companies, but also allows even the smallest retailers to advertise their services to a global audience. Specialised companies can flourish by harnessing the internet's immense reach to deliver to potential customers.

Of course, this growth has resulted in casualties in traditional (sometimes called 'bricks and mortar') retailing. High-street shops specialising in items such as books, DVDs, music and games have all lost business to online retailers. This has driven some shops out of business, but a number of them have also opened successful online stores. Online retailers have lower costs – they don't pay expensive high-street rents and can easily be based in countries with low tax regimes – and they can pass these savings on to customers. There are other concerns about these large online retailers, such as the employment terms and conditions of warehouse workers.

Although the low costs offered by online stores can be very attractive to customers, they might be counterbalanced to some extent by the less immediate and less tangible nature of the shopping experience. However, online retailers try to make up for this by providing a variety of services that reassure and inform their customers. One such service is the tracking of goods online – though this is a development that can produce its own frustrations, as illustrated by the cartoon in Figure 1.5. Having bought a new computer a few years ago and tracked it on its way from the factory in China to my home, I can identify with the feeling expressed! Interestingly, in 2015, Amazon set up 'bricks and mortar' shops near its headquarters in Seattle.

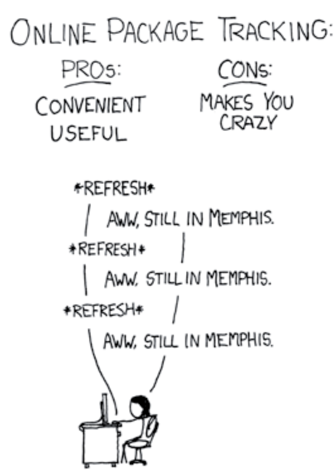


Figure 1.5 A light-hearted look at the pros and cons of online package tracking

Activity 1.5 (exploratory)

Add some more entries to the following table of advantages and disadvantages of online shopping. Can you think of ways in which online retailers may try to address the disadvantages?

	Advantages	Disadvantages
Buyer	More choice	Can't try goods (e.g. shoes, clothes) first
	Can often track goods	
Seller	No need for a physical shop	Reliant on delivery services

Comment

I thought of the following, but you may have come up with others.

	Advantages	Disadvantages
Buyer	More choice	Can't try goods (e.g. shoes, clothes) first
	Can often track goods	Harder to return goods if faulty
	Often cheaper than a local shop	Harder to get help and advice before or after buying
	Some items can be downloaded immediately after buying	May be worried about online fraud
Seller	No need for a physical shop	Reliant on delivery services
	Lower running costs	Likely to get more returned goods
	Don't need to keep everything in stock – can arrange delivery from suppliers	Need to respond to email or telephone queries
	Can supply music, books, software, etc. for download rather than having to supply a physical item	Some potential customers reluctant to buy online

Online retailers may try to address the disadvantages listed above in various ways. For instance, they might provide:

- free collection of returned items
- links to online reviews of products to help advise prospective buyers
- telephone helplines
- support forums to help customers before and after buying
- advice on how to shop safely online.

As well as direct retailing, other types of businesses have also moved online – auction sites such as eBay fall into this category. Some online businesses are less conventional, and as a result it's often harder to see how they find the money they need to survive. For example, there are many collaborative projects that produce free products including **software**, online encyclopedias and educational resources. These often rely on volunteers contributing their time, with money being provided by advertising, sponsorship or donations.

Work

Technology has changed the way that other businesses operate too. Greater quantities of information are exchanged between numerous locations over public and private networks. Vast amounts of data are stored on computers and accessed remotely from a variety of devices. Many companies now use **data centres** or centralised groups of networked computer servers that specialise in the storage, processing and distribution of large amounts of data.

Just as individuals buy from companies online, many companies now sell to each other online as well, for the same reasons of reduced cost and wider choice that attract individuals to online stores. Manufacturing tasks that used to take days can now be completed in minutes using computer-operated machine tools working in automated production lines.

The way we use technology has also affected our individual working lives. For example, the role of travel agents has changed as more people book their holidays directly from the vendor by going online. Some companies have responded by reducing their number of employees, while others have retrained their staff to provide more specialised services to their customers. More generally, many people working in an office environment are expected to learn how to use new software applications in order to do their jobs.

In this module, you will explore some of the roles that have emerged within these industries. You will also consider the job market and how technology has resulted in changes to the recruitment process. If you are studying TM111 to improve your qualifications and progress in your career, you should find this topic particularly relevant to you.

1.2.2 Communities, information and entertainment

Communities

As well as revolutionising the commercial world, the internet has had an enormous impact on the way we communicate. While there are still people in many parts of the world who do not have internet access, many of us have access at home or at work. As a result, we have the opportunity to communicate with others using email, instant messaging and **online discussion groups** (in online **forums**). Existing communities have created new ways of communicating, and new online communities have developed.

Social networking plays an increasingly significant role in the lives of many people.

The recent pace of communication change has amazed many of us, and there is no reason to think it will slow down. The cartoon map in Figure 1.6 illustrates the relative size and significance of online communities. The first version of this map appeared on the web in early 2007 when MySpace was the most important social networking site and the micro-blogging site, Twitter, didn't exist. Over time, new online communities may appear and grow and the current ones may decline, or others which were declining may increase in size and importance.

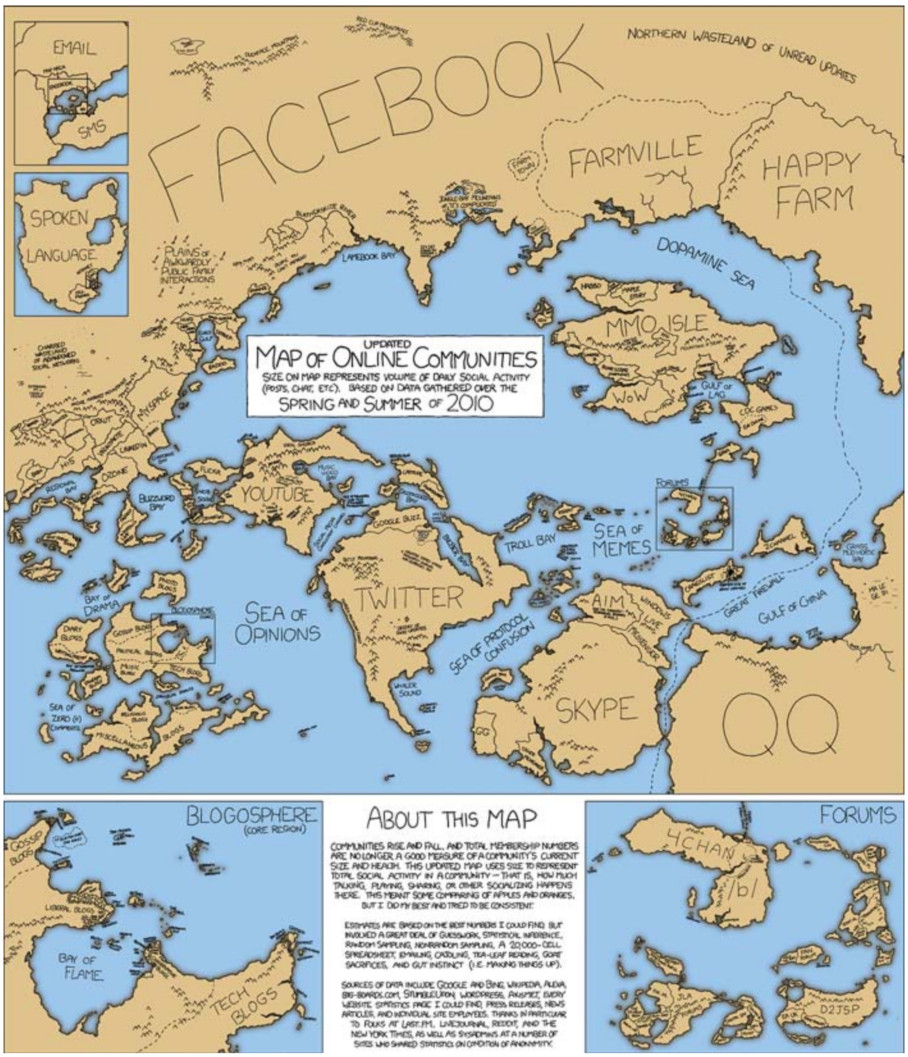


Figure 1.6 One view of the world's online communities in 2010

Information

The internet has had a huge impact on the availability of information of all kinds. Material on the web reflects widely differing viewpoints, from official news bulletins to unofficial rumours, and comes from widely differing sources, from commercial megastores to community groups. Since no individual government, company or person has control over it, the internet has paved the way to unfettered publishing of information of all kinds, raising questions about the authority and regulation of this information. As you will learn in your studies, some governments try to exert control over the information their citizens can access and create, with varying degrees of success.

The widely used online encyclopedia, Wikipedia, for example, contains millions of articles, many of high-quality scholarship – but there are also articles that are of suspect quality and may even be deliberately misleading. Its greatest strength is also its greatest weakness in that anyone with an internet connection can edit its pages. The encyclopedia itself advises users to treat with caution any one single source (whatever the media), or multiple works that derive from a single source.

The ever-increasing volume of online information available on the web means it is important to think critically about what you find, especially if you are going to use it for study or work purposes. You will be learning about how to find and evaluate information online throughout your study of TM111 as these are important digital and information literacy (DIL) skills. The Library website has useful resources and activities that could help you evaluate whether the information you come across during your studies is up to date and reliable.

Entertainment

The world of entertainment is constantly evolving as new ways of creating and distributing the media we watch and listen to are developed. Digital broadcasting has changed the way we experience television and radio, with increasingly interactive and participative programmes. Digital cameras, printers and scanners, together with desktop publishing and photo-editing software, enable greater numbers of people to experiment with image production, while online image- and video-sharing sites allow anyone with access to a relatively basic mobile phone or digital camera to share photos and videos with the rest of the online world. New digital technologies have also been at the forefront of changes in the production and distribution of music, and computer gaming has developed hand in hand with the evolution of graphical interfaces.

However, our increased exposure to digital entertainment has resulted in increased conflict between the rights of the consumer and the rights of the producer of the media. It is now much easier for the products of the media industries established during the twentieth century – film, music and so on – to be illegally copied and distributed in a form that is indistinguishable from the original. **Copyright** holders are taking steps to prevent this by developing a range of **digital rights management (DRM)** techniques that make it much harder to create copies, as well as by trying to persuade users of the benefits of the original product. Such attempts at persuasion can look very threatening, as I noticed on a recently purchased CD that has the following printed on the back cover:

> FBI Anti-piracy Warning:
>
> Unauthorized copying is punishable under federal law.

Several questions came to mind when I read this. Does that mean I can't legally put the music onto my MP3 player? Copying from one format to another is legal in the USA but, although the law is under review, it is not legal in the UK at present (although people in the UK are not deterred from doing it). Will I be in trouble only if I distribute copies of the CD to other people? Making copies and distributing them to friends is illegal on both sides of the Atlantic. These are all valid concerns that demonstrate some of the problems surrounding the use of copyrighted material.

Copyright is very much a 'live' issue in the UK and the US at present. There are many other issues arising from this, and it is very easy to make the digital future sound bleak. You have probably heard predictions to the effect that illegally copying media and making information freely available on the web puts whole businesses and hundreds of thousands of jobs in the established media industries at risk. However, as in other areas of the digital world, there are also opportunities for these businesses if they can adapt to the new environment and modify their business models to survive and grow in different directions.

Activity 1.6 (exploratory)

Can you think of any other problems connected with the growth of digital entertainment?

Comment

There are many possibilities. I thought of the fact that the advent of digital television affected even those who didn't particularly welcome it – across the world, analogue signals are gradually being switched off as new digital signals are introduced, as has happened in the UK. This meant that people had to buy new digital televisions as the old analogue versions become obsolete – quite an expensive business! At the time of writing, the government plans to switch off analogue radio soon, which means that people may need to upgrade to DAB (digital audio broadcasting) radio equipment.

You might also have thought of more technical problems, such as how to transmit the large quantities of digital data required for some forms of entertainment – video, for example – in an acceptable time and retaining acceptable quality.

1.2.3 Public information and services

Public bodies such as governments and transport agencies are increasingly providing services online, allowing us to organise various aspects of our daily lives more easily. These services range from simple information displays, which let us check things such as weather forecasts and transport timetables, to interactive sites that allow us to make bookings or queries.

In many parts of the world, medical records are increasingly moving away from paper and X-ray film towards becoming completely digital. This has several advantages, especially in allowing patient records to be easily shared between departments within a hospital, and sometimes more widely with doctors' surgeries and other health workers. In most parts of the UK, doctors and nurses walk around wards with tablet computers and other forms of handheld devices, instead of files and clipboards. In remote rural areas of some countries, doctors can make use of computer networks or even mobile phones to make a diagnosis if they are unable to see the patient in person. However, this is by no means universal, and even where such facilities exist they aren't always available.

Information for travellers is also increasingly being made available digitally: for example, live online updates on road congestion and public transport, and arrivals information in stations and airports. Similarly, most people book plane journeys online – in fact, some airlines now only accept online bookings and will only issue electronic tickets. Many of them require, or strongly encourage, passengers to check in online as well. You can track the locations of aeroplanes in real time using apps such as FlightAware and Flightradar24.

In addition, many countries provide online access to at least some of their government services. For example, you might be able to renew or apply for a passport, book a driving test, claim benefits, or fill in your tax return online. Local authorities also provide digital information services – usually you are able to reserve or renew a library book online, for instance – and there are numerous opportunities to learn online, as you will be doing on TM111.

Changing internet use over time

The Office for National Statistics has been collecting statistics on the use of the internet since 1998 and has produced an annual report since 2006 based on surveys of people in the UK.

The most recent survey at the time of writing shows a huge increase in the use of the internet and social media since their first annual report. These are the key findings:

- The internet was used daily or almost daily by 82% of adults (41.8 million) in Great Britain in 2016, compared with 78% (39.3 million) in 2015 and 35% (16.2 million) in 2006.
- In 2016, 70% of adults accessed the internet 'on the go' using a mobile phone or smartphone, up from 66% in 2015 and nearly double the 2011 estimate of 36%.

- Analysis of the use of smart TVs shows that 21% of adults used them to connect to the internet in 2016.
- In 2016, 77% of adults bought goods or services online, up from 53% in 2008, but only up 1 percentage point from 2015.
- Clothes or sports goods were purchased by 54% of adults, making them the most popular online purchase in 2016.
- In 2016, 89% of households in Great Britain (23.7 million) had internet access, an increase from 86% in 2015 and 57% in 2006.

(Source: Office for National Statistics, 2016)

Activity 1.7 (exploratory)

The findings from this survey come from a very reliable source, the Office for National Statistics. Do any of the findings surprise you?

Comment

I was surprised that 89% of the population have internet access (the last bullet point) because this suggests that 11% of households still do not have access. As this section has shown, so much of our world is digital. This raises questions about inequalities in terms of access to goods and services including public services. The inclusion of statistics on the use of smart TVs suggests that they are becoming a useful way of getting online.

1.2.4 Security and risk

The twentieth century saw a dramatic change in the role of the state in many countries. During most of the nineteenth century, an individual might only have come into contact with the state for the purposes of taxation, marriage and death; at the end of that century and the beginning of the next, however, a series of social revolutions saw the state becoming involved in our healthcare, pensions and education. Unsurprisingly, each of these developments was accompanied by a significant increase in the amount of personal information stored about every one of us. Computer technologies were developed especially to serve the enormous projects involved; IBM became a highly successful company due to its work on censuses in the USA and Europe, whilst the world's first business computer, LEO, was used for a variety of tasks including the calculation of tax tables for the UK Treasury in the 1950s.

With the vast amount of personal information being held about us in various places, it is becoming increasingly important for us to be able to prove our identities – not just for travel, but for other activities such as purchasing expensive or restricted items, paying bills and opening bank accounts. The UK is unusual in Europe in that (at the time of writing) it does not have a compulsory identity card system, despite the fact that identity cards were put in place during both world wars. However, effectively, driving licences, passports and some types of travel pass have become forms of identity card. In several countries, identity card or passport schemes are being upgraded with new **biometric** technologies such as iris or face recognition, which claim to identify individuals uniquely.

As well as the personal information that we know about, there may also exist information about us of which we are unaware. Since the terrorist attacks on the USA in 2001, much of the Western world has become far more security-conscious, and governments and companies alike have developed and deployed technological countermeasures. These range from smart video surveillance systems that can identify an individual in a crowd and track his or her movements, through to the biometric technologies mentioned above, to the searching of databases for suspicious activity.

Activity 1.8 (exploratory)

Can you recall an occasion when you have been personally aware of technological security measures?

Comment

My own experience of security measures at airports involves checking in online and giving details of my own passport and that of my travelling companion several weeks before travelling to mainland Europe. A colleague who visited the USA recently went through a range of airport security screenings. In addition to having his belongings checked, he was photographed at least twice (as well as being under almost constant video surveillance in the airports), had his fingerprints scanned electronically, and was required to fill in numerous online and paper forms.

The promise is that such technologies will make us safer, but could they turn the world we live in into a society strangely reminiscent of the nightmare vision of ‘Big Brother’ contained in George Orwell’s novel *1984*?

As you’ve seen so far in this section, there is plenty of opportunity for digital information about each of us to be created. Some of this information we might intentionally give out ourselves – on social networking sites, for example. Other information about us may, as described above, be gathered more surreptitiously by various agencies. In general, we have little control over how digital information about us is used or who receives it. We might assume that information gathered legally by a government agency, for instance, will be handled appropriately and used only for our benefit; yet there have been many examples of governments and private organisations ‘losing’ confidential data by storing or transferring it insecurely. For example, in 2015 the communications company, TalkTalk, was forced to issue a warning to its customers that their data had been hacked and that criminals were using the stolen information to trick people into handing over their bank details.

1.2.5 Communicating on the move

Advances in digital technology have, in a very short space of time, revolutionised the way many of us live our lives. Nowhere is this more evident than in our ability to communicate as we travel. When I first moved away from home, communicating with my parents involved a shared payphone in the hallway or a telephone box at the corner of the street. Nowadays I have a mobile phone with which I can send and pick up emails quickly and cheaply wherever I am. As well as email, my phone enables me to communicate in several other ways – instant messaging and social networking, for example, not to mention voice calls. I can also use the built-in **global positioning system (GPS)** to find out where I am, and can even plot my location on a website to let my friends and family see where I am (or at least see where my phone is).

1.2.6 Conclusion

In this section, you've learned about some of the key aspects of an information society, social networking being one of them. In the next section, I'll outline some ways in which you can participate in online discussions effectively.

1.3 Participating in a digital world

Previously in this part, I've mentioned the rise of social networking – one aspect of our increasingly digital world. You may already communicate online to some extent in your daily life, and while studying TM111 you'll certainly do so as part of an online community of tutors and students. In particular, you'll participate in your online tutor group forum, whose members are your tutor and all the students he or she has been allocated. In a later part, you will look again at the phenomenon of online communication in a variety of forms, and analyse it in some depth.

When we have a face-to-face conversation, we don't just rely on the spoken words to establish the other person's meaning; subconsciously, we are also monitoring the tone of their voice, their facial expression and their body language. Telephone conversations are a little more ambiguous because we can no longer see the other person; email and online discussions are harder again, since all we have is text. It is extremely easy to misinterpret words on a page, so the writer must take great care before pressing the button that sends their message to the world.

What follows in this section is a quick guide to good practice in contributing to online discussions. It should help you to work and socialise with others online – in your studies, your social life and your working life. Much of the content is summed up by the familiar tenet known as the Golden Rule: a concept common to many ethical codes, which simply states that we should treat others as we would want them to treat us. Just as this Golden Rule is relevant to good manners – 'etiquette' – when talking face to face, so it is relevant to online communication. To help us apply it, it has been developed into guidelines for online behaviour called 'net etiquette' or, more commonly, **netiquette**.

Netiquette is intended to make us all think about how we behave online and to make us aware of the effect our words could have on others reading them. These guidelines aren't hard-and-fast commands that you must remember and obey, and netiquette does not encompass every situation you may find yourself in. In fact, it's perfectly possible to obey all the guidance below and still annoy someone. However, netiquette provides a good foundation for your participation in online discussions.

Most of what follows is common sense and good manners. Some of it may be familiar to you, but please take time to read it – and do so especially carefully if this is your first Open University module or if you don't have much experience of using online discussion groups to work with others. There's a big difference between working in an online community and socialising online, so even if you are experienced at the latter, you should find the following material useful.

1.3.1 Netiquette: respecting others online

When children, we quickly learn many rules about how to interact with other people. Some of these rules are common sense, such as 'don't interrupt a speaker' and 'say please and thank you', and are necessary if we are to reduce the likelihood of arguments or causing offence.

Thank, acknowledge and support people

People can't see you nod, smile or frown as you read their messages. If they get no response, they may feel ignored and be discouraged from contributing further. Why not send a short reply to keep the conversation going? This can make a big difference in a small group setting such as a tutor group forum. However, do bear in mind that in a large, busy forum, too many messages like this could be a nuisance.

Acknowledge before differing

Before you disagree with someone, try to summarise the other person's point in your own words. Then they know you are trying to understand them and will be more likely to take your view seriously. Otherwise, you risk talking at each other rather than with each other. You should also recognise that other people are entitled to their point of view, even if you consider them to be entirely wrong.

Make clear your perspective

Try to speak personally. That means avoiding statements like 'This is the way it is …' or 'It is a fact that …'. These sound dogmatic and leave no room for anyone else's perspective. Why not start by saying 'I think …' or 'I feel …'? If you are presenting someone else's views, then say so, perhaps by using a quotation and acknowledgement.

Other cues

Emotions can be easily misunderstood when you can't see faces or body language. People may not realise you are joking, and irony and satire are easily missed – all good reasons to think before you send a message. To compensate for these restrictions, early internet users came up with the idea of the smiley face – :) or :-) – which then grew into a whole family of **emoticons**.

Small digital images called **emojis** are commonly used in electronic communications, such as text messages and emails, to represent ideas and emotions. Incidentally, the word 'emoji' comes from the Japanese term for 'picture characters' so, although the word is similar to 'emoticon', this is a coincidence. Examples of emojis are shown in Figure 1.7 below.

Figure 1.7 Examples of emojis

Remember that the systems upon which many forums are based only support plain text, so you can't always rely on emojis or fonts and colours to add meaning. Even if you are using a forum that allows so-called 'rich text', it's possible that other users will be picking up messages as plain-text emails or as text-message alerts on their mobile phones and will not see your formatting. AND DON'T WRITE IN CAPITAL LETTERS – IT WILL COME OVER AS SHOUTING!

If you read something that offends or upsets you, it is very tempting to dash off a reply immediately. However, messages written in the heat of the moment can often cause offence themselves. It's much better to save your message as a draft and take a break or sleep on it. That gives you a chance to come back to your message when you're feeling calmer and ask yourself 'how would I feel if someone sent that message to me?'. If you decide it will make things worse, then make sure you edit it before you send it.

The best advice is to try to be aware of your audience before you post. The internet is a global phenomenon; people from widely differing cultures and backgrounds may read what you write online, and what you find funny may be offensive to them. It may take time to work out what sort of 'audience' can be found in a particular forum; some are very permissive and allow almost any sort of behaviour, while most (like those at The Open University) will not tolerate bad behaviour or abuse.

Activity 1.9 (exploratory)

Look at the email exchange shown in Figure 1.8. What emotions are being expressed through smileys and typography? Would Jon and Sue still be on speaking terms if they hadn't used these devices?

Today at 13:00 J

Jon
To: Sue
Files

Dear Sue,
I've messed up by overwriting a file you've just edited.
I think I've mended the damage, but could you check I did it right?
Jon

Today at 13:05 S

Sue
To: Jon
Re: Files

Aaaarrrgggghhhhhh. Don't do this to me. ;o)

Today at 13:12 J

Jon
To: Sue
Re: Files

Sorry.

Today at 13:20 S

Sue
To: Jon
Re: Files

:o))))

The page looks OK. No harm done.

I had a meeting at 8.15 yes 8.15 today so forgive me if I'm a bit grumpy.

Sue

Figure 1.8 Email exchange between Jon and Sue

Comment

In Sue's first reply to Jon she expresses her frustration by typing 'Aaaarrrgggghhhhhh', but she ends that message with a winking smiley. Jon's reply then says 'sorry' in a very small voice! Sue's final reply starts with a happy smiley to show that everything's OK. She uses a large font when she mentions the annoyingly early meeting time.

I feel sure that Jon and Sue would still be friendly after this email exchange. But I have seen email exchanges between colleagues that had the opposite effect, when the participants didn't take care about how they expressed themselves in their messages.

Moderation of forums

Moderators are forum participants responsible for keeping order. They may have capabilities within the forum greater than those of other participants; for example, they can sometimes add new participants and suspend people who are abusive. They also work to keep the discussions friendly and relevant to the forum. A forum with a moderator is said to be *moderated*. All Open University forums are moderated, and your tutor group forum will be moderated by your tutor. Forums without moderators are *unmoderated* and are generally places where newcomers should tread very carefully. Moderators tend to introduce themselves early on in forum discussions, so it's usually clear whether a forum is moderated or not.

Some other advice

Other useful points to note when you are using a forum are:

- Keep to the subject, and pick the right forum for your contribution.
- Before you write a message, check any rules about what is and is not considered acceptable in the forum. Many discussion forums have rules, aside from netiquette, about things such as links to commercial sites.
- Take a little time to use the forum's search facilities to see if your question or topic has already been discussed or covered in a set of frequently asked questions (FAQs). If it has, you should at least scan the existing messages to see if your points have been addressed.
- Don't feel you have to post immediately. Take your time to see what is being discussed and get a feel for the group you're joining. This very sensible behaviour has the unfortunate name of *lurking*, but is quite acceptable online. If you want to post, many discussion groups have a forum devoted to new users where they can introduce themselves to other readers. These are always good places to get started.
- Try to keep your messages short and to the point. People don't want to read long, rambling messages, especially if they can't work out what response you're looking for.
- Write a concise subject line (title) for your message – people often won't spend time reading messages unless the subject line looks relevant.

- Keep to one subject (topic of discussion) per message. If you want to cover another subject, do it in another message.
- When replying to a message, quoting part of that earlier message can be helpful so that readers can easily see what you are referring to. Add your response *after* the quoted material, not before it. And keep your quotation short and to the point, otherwise the resulting messages will get longer and longer.
- If you ask a question and it is answered, thank the person who responded. It's not only polite, it also shows that the discussion has come to an end.
- If you've reached a point where you disagree with someone and neither of you is going to change your opinion (Figure 1.9), realise the conversation is over, agree to disagree, and move on.

Figure 1.9 Online discussions can become addictive!

1.3.2 Studying and working online

In this section, we will explore some of the online support and resources that are available to you in this module, your future studies and beyond.

Activity 1.10 (exploratory)

Firstly, on the module website, you will find a link to a quiz on netiquette. Use this quiz to test your understanding of what you have studied in the previous section. Then go on to the next activity, which also requires you to be online.

Your tutor group forum

This block is called *The digital world*, and so far in this part and in the TM111 Guide, I have explained a little about this module and introduced some of the technologies and issues we will be exploring more deeply in this module. Now it's time for you to tell other people a bit about yourself. You are a member of a tutor group that consists of several other students and your tutor. You may not have had the chance for a face-to-face meeting, so this is your opportunity to introduce yourself.

Activity 1.11 (exploratory)

If you haven't already done so, post a short message to your tutor group forum, introducing yourself to your fellow students and your tutor. If your tutor group forum is not yet open, you can prepare your posting in advance.

There are no hard-and-fast rules about what to say – all of us have different feelings about what we like to share with others. Some of us will say anything to anyone, some of us are more reticent. Only say what you're comfortable with. Don't worry if you're feeling a little shy – you won't be the only one.

A good start might be telling your colleagues your name and your town. Perhaps you'd like to say if you're employed (and if so, what your job is), if you've recently left school, if you're at home with children or with caring responsibilities, or retired? Is this your first Open University module? How about your knowledge of computers and information technology? Are you a novice in the field? What do you expect to get out of TM111 and why did you choose it? These are just examples – as long as you follow netiquette, it's up to you!

Your digital identity

As we live more of our lives online, so it becomes more important for us to be aware of and, as far as possible, try to manage the information that others can access about us online.

Activity 1.12 (exploratory)

I now want you to explore some library resources on managing your digital identity. Go to the module website and follow the link for this activity.

Comment

You'll learn more about this in a later part but, as a general guideline, before you place any kind of information about yourself on the web you should think about the impression you would like potential employers, new friends, your parents or your children to get of you if they searched for you online. Stories of people being disciplined or even losing their jobs as a result of inappropriate comments or photographs on their blogs or social networking sites show that this isn't a hypothetical situation. For some employers, checking the information that is available online about job applicants is as much a part of the selection process as taking up references.

Activity 1.13 (exploratory)

Enter your name into a search engine and review what you find.

(a) What is revealed about you?

(b) What is revealed about other people with the same name as you? If you have a common name, then it may be hard to tell the difference between you and others with the same name. However, you should be able to get an impression – and possibly even some photographs – of a few individuals who share your name. If you can't find anyone with your name, try looking for the name of a family member or friend. Make some notes on one or more of the people you find.

Comment

(a) My name (Elaine Thomas) is fairly common, so most of the links I found do not refer to me. There was one link to my Open University page, which links to other information about my work. However, when I added 'Open University' to my search, I found many links to presentations I have made at conferences and papers I have written, as well as links to the OU website. Although there are dozens of people called 'Elaine Thomas' on Facebook, I didn't find a link to my Facebook page because the privacy settings mean that it isn't publicly available.

(b) There were several references to people who share my name. These people include an American actor, a German musician, a lecturer in Biochemistry and someone who makes celebration cakes.

This so-called 'ego surfing' or 'vanity surfing' is an interesting thing to do from time to time. Overall, I haven't found anything unpleasant or embarrassing, but I'm old enough for any youthful indiscretions to have happened long before they could have been recorded online. I also have a range of usernames that don't resemble my real name and so aren't easily connected to me. They aren't used for anything nefarious, but I see no reason why I should make it easy for others to find out everything about me.

It's not something to get overly worried about, but it is worth being aware of the impression that others can get by following the trail of your online activities.

1.3.3 Ethical and legal considerations

In using a computer for communications, you have many rights of free expression, but you also have certain responsibilities to respect others. At this stage of your studies you need to be aware of privacy, confidentiality and copyright in relation to online communications. These are the main points you should be aware of:

- An email is generally considered to be equivalent to a private letter, and should not be quoted or forwarded to anyone else without the permission of the original sender. This can be particularly poorly observed in companies (even those whose employees are told to assume that all online communications are for the recipient's eyes only, unless otherwise stated).
- Besides the informal rules of netiquette, most forums have a code of conduct and conditions of use that govern acceptable behaviour. Your use of the online forums provided by The Open University is covered by the OU Computing Code of Conduct, further information about which is provided in the Computing Guide. You will usually find a forum's terms of use linked from its home page, or listed in the code of conduct that you are asked to agree to when you first register for an account.
- Considerations of copyright and **plagiarism** (cheating by using another person's work as if it were your own) apply to online discussions. If you are quoting something written by someone else, put it in quotation marks and acknowledge the source.
- Some forums are not wholly public – in which case, messages should not be copied outside the forum. The forum's terms of use may specify this.

1.3.4 Copyright

One of the reasons the web has grown so quickly, and one of its most fascinating aspects, is that almost anyone can publish almost anything on it. It is very easy to find information, images, audio and video files on the web, which you can then save and incorporate into your own material.

Copying is so easy that people often make the mistake of assuming that everything on the web is freely available. This is not the case: most material you will come across is likely to be covered by copyright law. This applies not only to online television programmes, music, photographs, books, and so on, but also to information – for example, in the form of online academic papers or simply in someone's blog.

Material not subject to copyright is said to belong in the **public domain** and can be used by anyone. Older works of art and literature, such as the works of Shakespeare and Beethoven, are in the public domain. However, individual printings, adaptations or recordings of those works are copyrighted to the publisher or performer. The situation can become very complicated because, for example, material may be in the public domain in the USA but still under copyright in the UK. As a result, it is always wise to assume that any third-party material (that is, material originating from someone else) is still protected by copyright unless you're sure it's in the public domain.

Copyright holders can prosecute individuals and organisations for infringing their rights; in recent years, music and film companies have sued individuals for very large sums of money. Below are some general points you should bear in mind.

- You should seek the author's permission if you wish to use any copyrighted material. Just because something is on the web does not mean it is freely available for you to use.
- Many websites have usage policies explaining how their material can be used. Some are more restrictive than others, so make sure you find and follow the relevant policies.
- Information published online may have been put there by someone who is not the copyright holder.
- You can make use of copyright print material for your own study purposes: the generally accepted guideline from copyright legislation is that you can download and print or photocopy up to 5% or a whole chapter of any one book without seeking permission, but you must give a full reference to show where it has come from.

This may all seem rather intimidating, but it's not quite as bad as you might fear – fortunately, copyright law makes some concessions to students. As a result, you don't have to ask permission to use copyrighted material in answering an assignment question, although you must still include references. However, if you want to reuse the same material for any other purpose at a later date, normal copyright law applies and you must seek permission.

The Library website includes further information on copyright.

1.3.5 Good academic practice

One of the things TM111 encourages you to do is use the web as a resource in your study and your life generally. You'll use it in your assignments, and you should find it a great help in understanding and practising the things you learn.

However, using information found on the web in this way can cause problems unless you take a little care. When using material written by other people, you can quote their words (as you saw above), but good academic practice is that such quotations should always be *limited* and *acknowledged*. This applies whether you're quoting from the TM111 materials or from other sources such as websites, journals or newspapers.

It can be very tempting to copy and paste large chunks of text into your notes – and possibly then into your assignment answers – without giving a reference. However, that is very bad academic practice. It's far better to use quotations sparingly and to rewrite most of the material in your own words. This allows you to show that you've understood the material and it also helps you to remember it.

In addition, it's good academic practice to give a reference to the source of any third-party material you include in your own work. Not doing so is not only impolite, as you're failing to acknowledge the help that someone else's work has given you; it's regarded as plagiarism and is never acceptable.

TM111 will teach you the correct way to use the work of others and help you to establish good academic habits. As well as using information found on the web, you will be asked to collaborate with other students to create and share information. You'll also be encouraged to discuss aspects of the module in your tutor group forum. Both of these activities will help you to develop academic practices that will stand you in good stead in future studies, as well as being valuable skills more generally. You can probably guess that, just as with other third-party materials you make use of, it's important to give references to the contributions of other students when you include them in your own work; otherwise you could fall into the trap of plagiarising their work, which is definitely something to be avoided.

Unless you are specifically asked to do so in the TM111 materials, you should not pass your work on to other students, especially if it is part of an assignment. One good reason for this is that The Open University uses detection software that is very good at spotting plagiarism in assignments! If you're in any doubt about what is acceptable and what is not, then do ask your tutor. The boundary between collaboration and plagiarism is sometimes not clear, and your tutor will help you understand whether you are in danger of crossing it.

To help you with this, the OU Library's website provides a guide on how to reference sources of information correctly, including those you might find online. Links to that and to other useful sites on how to follow good academic practice and avoid plagiarism are on the module website.

1.3.6 Conclusion

In this section, I've discussed some of the ways in which you can make good use of the web, both for interacting with other people and for finding information. When working online, it is also important to consider how to protect yourself and your computer, and that's what I'll turn to next.

Online safety

1.4

The internet provides many ways for people to get in touch with each other, but this ease of contact can have downsides for the unwary. It can expose internet users to the dangers of malicious software, to unsolicited and nuisance emails, and to a variety of hoaxes. In this section, I'll describe some of these problems and suggest how you can protect yourself from them in order to keep your personal data safe.

1.4.1 Malware

Activity 1.14 (exploratory)

What do you understand by the term 'malware'? Write down a sentence to explain this term.

Comment

Software designed to cause damage is known as **malware**. There are several types of malware, four of which are described below. However, be aware that as malware evolves to avoid detection, the boundaries between the different categories are tending to blur.

Types of malware

The best-known type of malware is probably the **virus**. This is a piece of software that has been written to attack software on your computer, often with the specific intention of causing harm – deleting files, for example. A virus attaches itself to other software on your computer and activates when that software is run. Viruses are so called because they are designed to spread quickly and easily from one computer to another via internet connections or external storage devices such as memory sticks.

Another type of malware is the **worm**. This is a piece of malicious software that runs 'in the background', doing some damage to your computer even though you may not realise it is running. Worms can make copies of themselves, and those copies can spread via an internet connection. A worm typically consumes resources by running on a computer; in a major attack, all of a computer's processing resources could be used in running the worm and its copies.

The **trojan** is a digital equivalent of the legendary wooden horse that smuggled Greek soldiers into Troy. It appears to be legitimate software, such as a screensaver, but behind the scenes it is causing damage – perhaps allowing someone else to gain control of the computer, copying personal information, deleting information, or using email software to pass itself on to other computers.

Finally, **ransomware** is a type of malware that prevents a user from accessing their computer either by locking the computer screen or by blocking access to computer files until a 'ransom' is paid. Typically, this form of malware is used to extort money from the user to 'ransom' the computer, but it could also be something relatively minor, such as being obliged to complete a survey.

Protecting your computer

There are three main ways to protect your computer against malware.

1 Ensure that your computer has the latest **patch** from the producer of your **operating system (OS)**. Microsoft, Apple and other producers frequently issue patches for their products.
2 Make sure other software is kept up to date – Adobe Reader, Flash, Java and web browsers (such as Mozilla Firefox, Google Chrome, Opera, Safari, etc.), to name just a few. As new malware is discovered, so new versions of software are released that guard against it.
3 Install **anti-virus software** and keep it up to date. Anti-virus software catches a very high percentage of malware, but only if the version on your computer is regularly updated. Remember that if you don't use Windows, it is still possible to pass on files infected with malware to Windows users. That's why the main job of anti-virus software for Apple's Macintosh operating system, macOS, is to check files for things that could infect Windows machines.

In addition, you can use a piece of software called a **firewall**. This tries to stop unauthorised access to your computer without impeding your own authorised online access. There may be a firewall built into your computer's operating system; others may be present in the hardware that connects your computer to the internet.

As well as the technical protections described above, you should protect yourself by using anti-virus software to scan any files you receive before you open them. This should include:

- files you download from the web
- files given to you on removable media such as a CD or memory stick
- files attached to emails.

Bear in mind that no reputable software company sends unsolicited email messages with attachments, claiming to be giving you an update.

1.4.2 Spam

Spam is the general term for unsolicited emails or text messages sent to large numbers of people. They could be hoax messages designed to mislead, or they could be used to advertise a product.

In terms of advertising, spam is similar to the marketing leaflets and letters that drop through your letterbox at home. However, this paper mail is subject to legislation that tightly controls the range of products and services being offered. The equivalent legislation does not yet exist in the electronic world, although new laws are being introduced. For example, in Europe, the EU 'Directive on Privacy and Electronic Communications' came into force in the latter part of 2003 and covers both email and SMS spam. In the USA, the federal law 'Controlling the Assault of Non-Solicited Pornography and Marketing' (CAN-SPAM) took effect in January 2004. Though such national legislation is intended to limit the volume of spam email, in practice this is a very difficult task because the internet crosses national borders. Spam can be sent from one country to another, and countries that have legislation find it hard to enforce their rules in countries that do not.

Spam email can be sent only if the *spammer* (the person initiating the spam) has a collection of email addresses to send to. A 'random email generator' is a computer **program** that creates addresses by combining common surnames and online email account names, e.g. gmail and yahoo. Other ways to 'harvest' email addresses include:

- accessing company databases
- searching websites
- searching online discussion groups
- including links in images within emails, which when clicked by the recipient inform the spammer that the message has been opened
- infecting unprotected computers with malicious software to look for addresses.

Spammers may harvest vast numbers of email addresses, but not immediately know whether a particular email address is 'live' (actually in use) – it could be that the original owner of the address no longer uses it. So beware of spam emails that appear to give you the option to unsubscribe from a mailing list (very often by offering a web link to click on). If you select this option, this will verify to the spammers that your email address is live; they can then continue to send you spam, or even sell your email address to other spammers.

Below are some guidelines for minimising the spam you receive.

- Don't reply to spam emails.
- Don't use the unsubscribe option in response to unrequested emails.
- Don't reveal your email address unless you want to receive mail from a particular source.
- Don't post your email address on a website.
- Create a new email address when registering on websites or joining discussion groups. You could also use a spare one that you're happy to abandon if necessary (e.g. a web-based email account such as those offered by Yahoo, Microsoft, Google).
- Set your email software to filter out unwanted messages. Most are equipped with 'junk mail' filters that can be set to identify and remove

spam messages as they arrive in your inbox. Additionally, your **internet service provider (ISP)** may filter incoming mail for spam before it even reaches your inbox.

- Ensure that all other users of your computer follow the above guidance.

1.4.3 Hoaxes

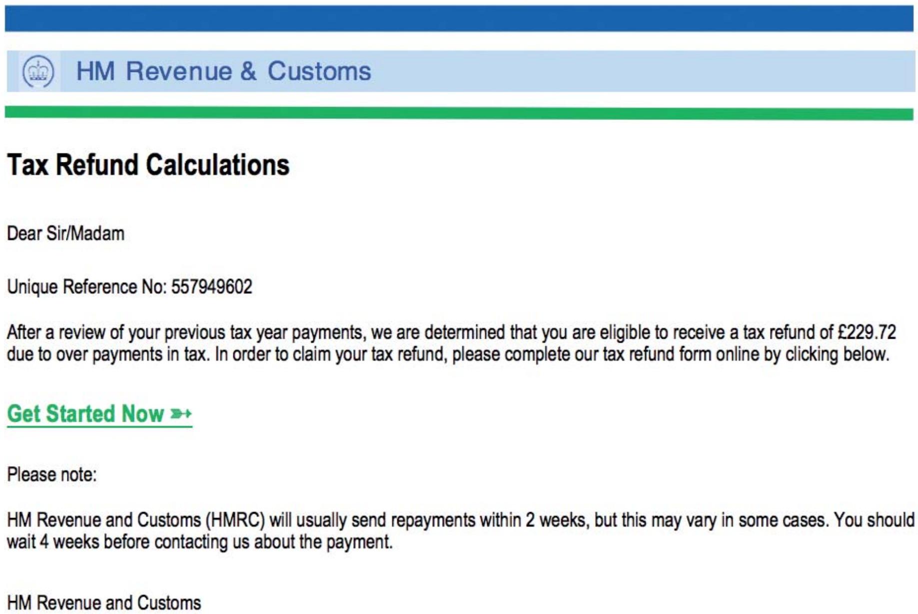
A hoax message aims to mislead, often relying on the naivety of its recipients. Sometimes these are fairly benign; they may spread 'urban myths' or untrue news stories. A recent (2015) hoax message claims that Mark Zuckerberg, founder of Facebook, will give $45 million to a thousand Facebook users. To be selected for the chance of winning, the recipient just has to copy and paste the message into Facebook and tag five to ten of their friends.

Other hoaxes are more malicious; for example, emails that appear to come from a legitimate source advising people to delete particular files on their computers, which causes their computer to malfunction. Hoax messages can spread rapidly via email and forums, often passed on unwittingly by work colleagues, family, friends and even reputable online retailers. Unfortunately, users who fall for hoaxes can cause problems both for themselves and for others. A hoax can generate spam when it directs the recipient to pass on the message, cause files to be deleted unnecessarily and potentially harmfully (by directing the user to delete them), and generally cause panic.

Your best defence is to be sceptical! Most anti-virus software vendors maintain information on hoaxes on their websites, so you can check such sites if you suspect a hoax. Alternatively, you can use a search engine: by searching for significant terms contained within the hoax message, you may find reports on reputable websites such as that of the software security company, Sophos, or the Hoax-Slayer site.

1.4.4 Phishing

A particular kind of hoax message aims to persuade users to disclose private information such as their credit card details and PIN. This is called **phishing**. The recipient of the message may be directed to a hoax website where they are requested to part with their details. Figure 1.10 shows an example of an email that I received which supposedly came from HM Revenue & Customs (HMRC).


HM Revenue & Customs

Tax Refund Calculations

Dear Sir/Madam

Unique Reference No: 557949602

After a review of your previous tax year payments, we are determined that you are eligible to receive a tax refund of £229.72 due to over payments in tax. In order to claim your tax refund, please complete our tax refund form online by clicking below.

Get Started Now ➻

Please note:

HM Revenue and Customs (HMRC) will usually send repayments within 2 weeks, but this may vary in some cases. You should wait 4 weeks before contacting us about the payment.

HM Revenue and Customs

Figure 1.10 An attempt at phishing

Activity 1.15 (exploratory)

(a) Looking at the email in Figure 1.10, what warning signs are there that might alert you to the fact that this is an attempt at phishing?

(b) What simple checks could you carry out to determine whether the email was malware?

Comment

(a) At first glance the email seemed in order, but there were a few warning signs.

- The message is not addressed to the recipient (me) by name, although it mentions a supposedly unique reference number and a specific sum of money. This can indicate a message sent randomly to large numbers of email addresses.
- The wording of the message is a little strange, e.g. 'we are determined that you are eligible'. I think it unlikely that such errors would be present in genuine messages from HMRC.

(b) In the actual email, you can position your mouse pointer over the link ('Get Started Now' in the example in Figure 1.10) or long-press on touch screen devices, such as smartphones. Look at the web address that appears either as pop-up text or at the bottom of the message window. If it seems to be unrelated to the sender of the message, then you should be even more suspicious. You can also position the mouse pointer over the sender's email address to check who it is actually from. After checking HMRC's website, I discovered that they never send information about tax rebates or refunds by email, and the website shows examples of phishing attempts.

The above points convinced me not to follow the instructions in the message. Yet despite all these warning signs, some people do hand over their details in this way.

It's very important *not to click* on any links in these sorts of messages. Even if you don't enter your account details, making any response at all may confirm to the phisher that your email address is valid, leaving you open to further hoaxes and spam. Simply clicking on a link also risks your computer being infected with malware that could distribute the same message to all the email addresses – including those of your work colleagues, family and friends – stored on your computer.

After receiving the email shown in Figure 1.10, I searched online and found a reputable site (millersmiles.co.uk, but there are others) that contained the information shown in Figure 1.11.

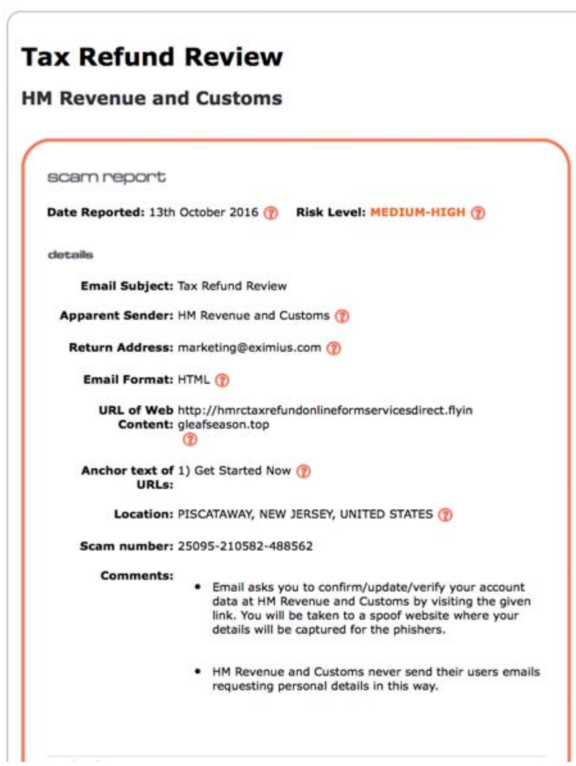


Figure 1.11 Information about a similar phishing attempt

Most phishing messages try to get you to provide some personal information (see Figure 1.12). Clearly those trying to get your online banking details are aiming to get access to your money. However, others could be trying to access your email, blog, instant messaging or online auction accounts (if you have any), then use these accounts to distribute more phishing emails.

A more focussed type of phishing, spear-phishing, uses clever tactics to target a specific individual or an organisation seeking access to sensitive information. Spear-phishing attempts are usually conducted for financial gain or to carry out espionage.

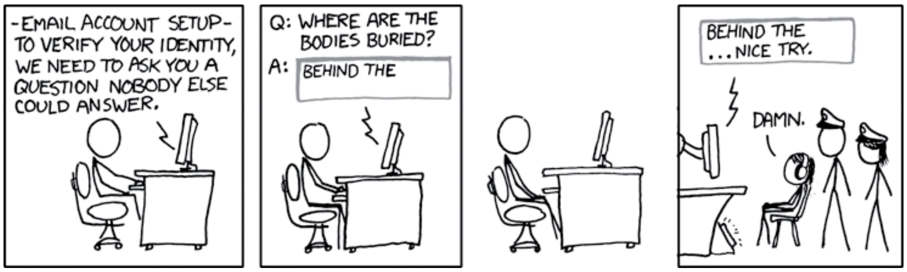


Figure 1.12 Phishing: fooling someone into giving away secret information

If you have a job that provides you with an email account, then you have probably also seen messages claiming to be from your company's IT department, asking you to enter your username and password into a website to reset your email account. I've certainly had them sent to my Open University email account. As with requests for you to disclose your financial details, you should be careful when disclosing your username and password. A phone call to the relevant people in the company should be sufficient to find out whether or not the request is genuine.

1.4.5 Conclusion

In this section, I've discussed ways in which you can protect your computer and yourself online, ranging from taking sensible precautions against malware to guarding against presenting information about yourself to the world that you might later regret.

Summary

In this part, I've introduced several aspects of our increasingly digital lives that you will be learning more about, and indeed experiencing yourself, during your study of TM111. You've heard about the ever more significant role that digital technologies are playing in our digital, networked society, in areas ranging from entertainment to public services. Along with the opportunities available in the online world, you've also been introduced to some of the potential problems and how to avoid them.

In Part 2 of this block, you'll learn more about the development of the computer. Before you move on to that, however, it is important that you take stock of what you've learned so far. Considering the main study points at the end of each part is a good way to check your progress.

After studying Block 1 Part 1 you should be aware of, and understand:

- the concept of a 'digital' world
- the role of information and communication technologies in a digital world
- the importance of managing your online identity for personal and work reasons
- the legal, ethical and safety issues involved in the digital world
- the importance of making and keeping notes as you study
- how to work collaboratively online applying netiquette principles.

And finally…

If you didn't get a chance to do the 'How digital is your life?' quiz earlier, then please do it as soon as you can. You should also check the study planner on the module website and ensure that you have investigated the links related to Block 1 Part 1 there. Otherwise, you're ready to start Part 2, which directly follows this part in the book.

Answers to self-assessment activities

Activity 1.4

(a) The 65+ age group shows the smallest increase in use of instant messaging of 9%.

(b) The 25–34 age group shows the largest increase in use of instant messaging of 42%.

(c) One reason for the increase in use is that smartphones have become much more ubiquitous. While they are still expensive, the handsets are cheaper and more intuitive to use and the ‘bundles’ of telephone and internet access have become more affordable.

References

Associated Press (2012) 'Mobile carriers face a future without text messaging as data apps take over', *The Guardian*, 1 March [Online]. Available at https://www.theguardian.com/technology/2012/mar/01/sms-pinger-whatsapp-messaging-mwc (Accessed 6 February 2017).

Evans, B. (2015) 'WhatsApp sails past SMS, but where does messaging go next?', *Benedict Evans*, 11 January [Blog]. Available at http://ben-evans.com/benedictevans/2015/1/11/whatsapp-sails-past-sms-but-where-does-messaging-go-next (Accessed 6 February 2017).

International Telecommunication Union (2006) *ITU Internet Report 2006: digital.life* [Online]. Available at www.itu.int/osg/spu/publications/digitalife (Accessed 6 February 2017).

Ofcom (2015) *Adults' Media Use and Attitudes: Report 2015* [Online]. Available at https://www.ofcom.org.uk/__data/assets/pdf_file/0014/82112/2015_adults_media_use_and_attitudes_report.pdf (Accessed 20 October 2016).

Ofcom (2016) *Telecommunications Market Data Update Q1 2016* [Online]. Available at https://www.ofcom.org.uk/research-and-data/telecoms-research/data-updates/q1-2016 (Accessed on 13 March 2017).

Office for National Statistics (2016) *Internet Access – Households and Individuals: 2016: What the Internet is Used For and Types of Purchases Made, by Adults (Aged 16 or Over).* Available at https://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/bulletins/internetaccesshouseholdsandindividuals/2016 (Accessed 6 February 2017).

Acknowledgements

The TM111 module team acknowledges the contribution of John Woodthorpe to the production of these materials.

Grateful acknowledgement is made to the following sources.

Figure 1.1. Hampshire County Council, Provided by Hampshire Cultural Trust

Figure 1.2. Taken from: http://media-cache-ec0.pinimg.com/736x/3f/6c/35/3f6c35695014f2741197107b7e536fa2.jpg

Figure 1.3 Jim Zuckerman/Alamy

Figure 1.4. Taken from: https://www.ofcom.org.uk/__data/assets/pdf_file/0024/83166/adults_media_use_and_attitudes_charts_-_section_3.pdf

Figure 1.5. xkcd.com This file is licensed under the Creative Commons Attribution-Noncommercial-ShareAlike Licence http://creativecommons.org/licenses/by-nc-sa/2.5/

Figure 1.6. xkcd.com This file is licensed under the Creative Commons Attribution-Noncommercial-ShareAlike Licence http://creativecommons.org/licenses/by-nc-sa/3.0/

Figure 1.9. www.xkcd.com This file is licensed under the Creative Commons Attribution-Noncommercial-ShareAlike Licence http://creativecommons.org/licenses/by-nc-sa/3.0/

Figure 1.11. Taken from: http://www.millersmiles.co.uk/email/tax-refund-review-hm-revenue-and-customs

Figure 1.12. xkcd.com This file is licensed under the Creative Commons Attribution-Noncommercial-ShareAlike Licence http://creativecommons.org/licenses/by-nc-sa/3.0/

Every effort has been made to contact copyright holders. If any have been inadvertently overlooked, the publishers will be pleased to make the necessary arrangements at the first opportunity.

Part 2 The evolving computer

by Elaine Thomas

Contents

Introduction

The first computers weighed several tonnes, occupied entire buildings and were very rare indeed – in the early 1950s there were fewer than one hundred of them in the world. The notion of portability – simply being able to carry a computer from place to place – as well as the vast range of current uses to which computers are put would have seemed very strange to the people who built and worked with those early machines. Yet today, as you saw in Block 1 Part 1, computers and information technologies are practically everywhere in the world around us, and the processing power of an entire 1950s computer can be found on a *silicon chip* that is hardly visible. In fact, at the time of writing, it is estimated that there are almost 2 billion personal computers in the world!

In this part of TM111 I will relate some of the history of computing, from its very early days to the present. In looking at this I hope you will see that as the role of computers has changed, so our perception of what a computer is and how it should be used has altered too. As part of this discussion I will introduce you to concepts of computing such as *ubiquity* and the *cloud*, and describe how computers and networks are perceived to be evolving into the future. I will also show you the binary number system which forms the basic building blocks from which computer data and programs are formed.

I hope this part will help you to see the relationship between our expectations of computers and their evolution, so that you are able confidently to answer questions such as 'What is a computer?' and 'What are (and were) they used for?'.

To start you thinking about the history of computers, I'd like you to think about your own experience of computers.

Activity 2.1 (exploratory)

What is the first computer that you remember using? It could be anything from the computer you are using to study TM111 to one you used at work or owned/shared/borrowed in the past. If possible, you should identify the differences between the computer you used then and the one you use today. Can you add anything about your experience – for example, how easy and intuitive it was to use?

Comment

Obviously, the type of computer that you first used will depend on your age. My first experience of using a computer was the Amstrad PCW in the late 1980s. As I recall, it had a floppy disk drive and there was no internal storage for programs or data. Therefore, I had to load the program from disk each time I used it, take out the program disk and then load the data disk to store my work, otherwise it would have been lost whenever I turned off the computer. Over the last ten years, however, the differences between computers are mainly in terms of speed, size and portability. Also, the way we interact with the computer has changed, e.g. touch screens may be used as well as, or instead of, keyboards.

As you study this part, you will see how the pace of change in Computing and IT is accelerating over time.

Study note 2.1 Numeracy

This part contains several numeracy activities. If it has been a while since you did any numerical calculations, then you may want to take some of the sections slowly, perhaps going over them more than once. Do take time to try the activities for yourself.

If you need more practice, the *Using numbers* booklet provided with your module materials contains further examples for you to try.

Computers, but not as we know them

2.1

I'm now going to take you on a short tour of the history of computing, identifying some of the trends that have emerged and highlighting a few of the more significant stages and developments in technology that have led us to where we are today.

At this stage, you may be wondering why you need to know anything about the historical development of computing. After all, to make a journey you only have to know where you're going, not where you've been. The answer is that, where computers are concerned, there is no obvious destination, because the technologies involved are – and always have been – in a state of rapid development. This continual development is due not only to our ability to create new technologies to address particular challenges, but also to our creativity and vision in applying emerging technologies in new ways. Studying a little of the history will enable you to stand back and take a more objective view, to see the paths of development and application and how they influence each other. It should also give you some idea of the incredible rate of development that has taken place over the last few decades, which in turn will give you a feel for some of the developments that may still be in store.

This section is not intended to be a definitive history of the computer; such a history would run to several volumes. However, after studying this section, you should be able to identify a few of the milestones in the development of computers – particularly in the development of the personal computer (PC), although, as you will learn later, it is far from being the most common form of computing device.

There is a general acceptance that, to date, four generations of computer can be identified. Although this isn't universally agreed, it is very convenient to be able to group computers in this way when describing their development, and so in this section I will look at the first three generations in turn. As you work your way through the following sections, make notes of the key points you identify.

2.1.1 The first generation

Computing has its origins in mechanical devices that were designed to help solve arithmetical problems. There are many working examples and designs stretching back over the last few hundred years, or even several thousand years, if you include such things as the Greek Antikythera clockwork mechanism (shown in Figure 2.1) and the abacus.

Figure 2.1 The Antikythera mechanism, part of an astronomical clock built *c.* 150–100 BC

Essentially, the answer to the question of what was the first computer largely depends on how you choose to define a computer. In TM111 the focus is on *programmable electronic computers*, which first emerged around the time of the Second World War (AD 1939–1945).

Programmable computers

What do I mean by programmable? You can think of a program as a set of instructions, such as:

- add two numbers
- then divide the result by 3.

The result will vary depending on which two numbers are inputted into the program, which is useful. For example, if I input the numbers 5 and 7, then the result given by the program is 4; if I input 15 and 6, then the output is 7. More significantly, I can change the set of instructions to do something different, such as:

- subtract one value from the other
- then multiply the result by 4.

This implies a level of versatility, since I can change the set of instructions (the program) in order to produce different results. It's this ability to change the instructions that makes a machine programmable.

As well as the term program when talking about computers, we also talk about **data**. In the examples above, the data are the inputted numbers and those outputted by the program. The word *data* is actually plural (*datum* refers to an item of data), but is often used in the singular, e.g. 'The data is …'. You will find both uses in TM111, depending on the context.

Electronics

At the heart of every modern computer are **electronic switches**, and each switch can be either on or off. These states, on and off, can be thought of as representing the digits 1 and 0: when a particular switch is on, it represents the digit 1; when off, it represents the digit 0. Multiple switches can be set to different combinations of on/off to represent different sequences of 0s and 1s. You will see later in this part that every number can be represented by a unique sequence of this kind; such a sequence is known as a **binary number**. This ability to represent numbers using switches forms the basis of the programming capabilities of computers. The key to making a good computer is the ability to make switches that can be operated quickly and reliably with a minimum of effort.

Colossus

Colossus was the name given to a series of machines created and used at Bletchley Park in England during the Second World War. These machines, which are fine examples of very early programmable electronic computers, were dedicated to breaking codes produced by the German cipher machines known as Lorenz machines.

As their name indicates, the Colossus machines were quite large (Figure 2.2). They used bulky, expensive **valves** as electronic switches. A Colossus machine was fed using paper tape with holes punched in it to represent data. This tape was read into the machine at a speedy 30 mph and the output was to an electric typewriter (effectively a printer).

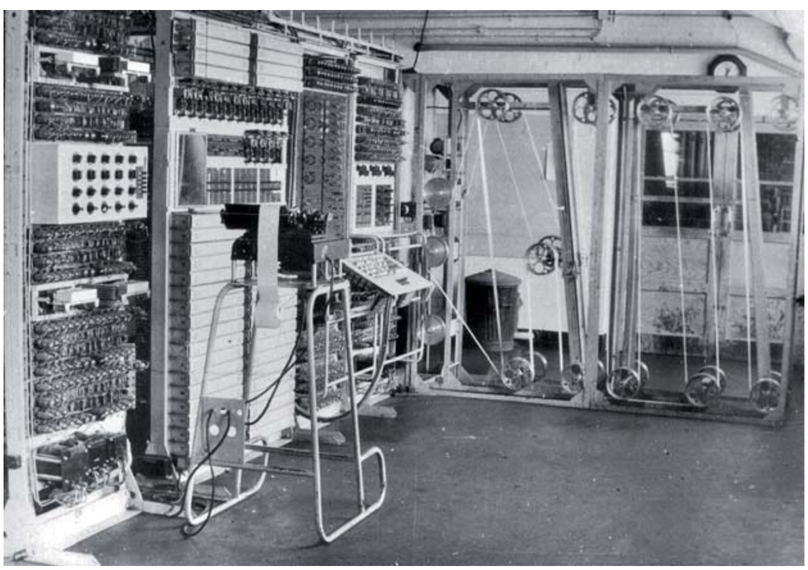


Figure 2.2 Colossus programmable electronic computer

Such was the secrecy surrounding the work done at Bletchley Park, in particular the deciphering of the Lorenz codes, that information relating to Colossus only entered the public domain in 1970.

There were many disadvantages to valve technology: valves had a limited life expectancy, they were large and they consumed a lot of electrical power. For a computer to be of any use, it needed to have thousands or even tens of thousands of valves. Thus the computers of the 1940s were unreliable, heavy and inefficient compared to the machines of today.

Box 2.1 Vacuum valves

Vacuum valves were used to control the electric current in first-generation computers. The current is passed between electrodes in an evacuated container (i.e. a vacuum). The valves are similar to light bulbs in appearance and behaviour. They also generated a lot of heat, and so frequently burned out and needed to be replaced.

ENIAC

The first *general-purpose* computer (remember that Colossus was a set of highly dedicated machines used for code-breaking) was probably ENIAC (Electronic Numerical Integrator and Computer). This was built at the University of Pennsylvania in the USA, again for military applications and again using valve technology. ENIAC weighed about 27 tonnes and occupied 63 square metres of floor space (it was 26 metres long). Figure 2.3 should give you an idea of its size.

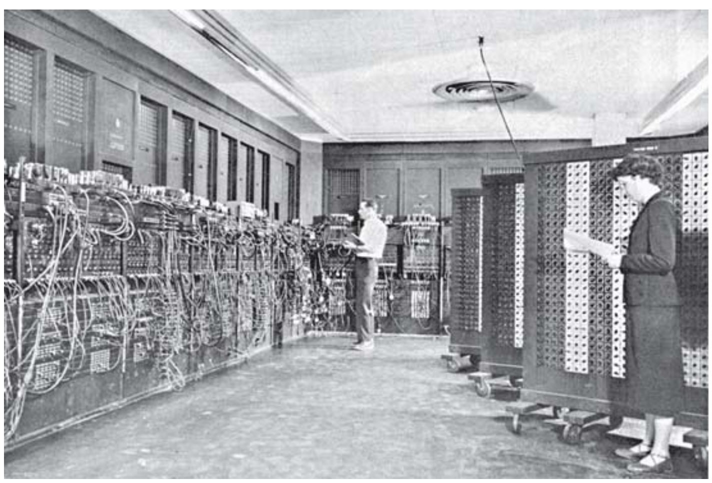


Figure 2.3 ENIAC

Like many of the early computers, ENIAC was primarily used to process lengthy calculations for the US military, particularly relating to the hydrogen bomb. It was programmed using plugs and wires, and required a team of operators to set it up and run it. Though it may seem odd to us now that a program was represented in this way, all a program does is establish an electronic circuit in which switches are turned on and off – so why not use plugs and wires to establish the circuit?

Of course, as you can probably imagine, ENIAC was cumbersome to use. Simply working out how to set up a program reputedly took weeks, and implementing it once it was set up took further days. Also, there was no way to save a program, so to return to the same calculation at some future date would require all the days of implementation to be repeated. In addition to these problems, ENIAC was initially very unreliable because of valve failures. It was under repair almost as frequently as it was working, although this improved as valves became more reliable.

Activity 2.2 (exploratory)

How have you made notes for this section so far? You may find it helpful to look at Figure 2.4, which is the spray diagram I created whilst working on the first part of this section. How does it compare with the notes you have taken? Have you identified similar points?

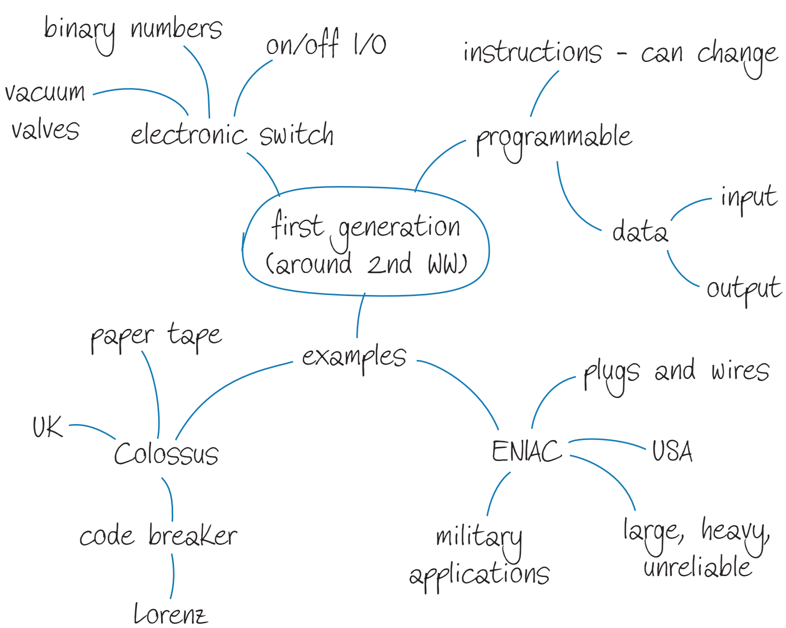


Figure 2.4 Spray diagram on the first part of Section 2.1

Comment

Although note-taking is a very individual activity, organising your notes into a spray diagram can help you to understand study material. Obviously a spray diagram won't fit in the margins of this printed text, so you'll have to use a separate piece of paper – maybe several pieces. You may have used another technique to make notes. Remember, you are taking notes as an active study strategy.

Storage of instructions

A major step forward came when computer designers realised that they could store instructions in the same way that data was stored: using a number of switches, each set to either on or off. This introduced the possibility of being able to store and recall sets of instructions, rather than wiring them in, as was the case with ENIAC.

By the early 1950s, computers were being sold for business use, often for tasks such as payroll calculations and stock control. Although they were still used to carry out calculations, there was a shift towards doing large numbers of simple calculations rather than the lengthy and complex calculations required by the military. For example, a UNIVAC I computer was used to predict the outcome of the 1952 US presidential elections based on statistical analysis of early returns.

Power consumption

At this point, it's worth pausing to consider the power consumption of electrical equipment. Power consumption affects different people and organisations in different ways. For some, the potential environmental impacts of power consumption are a key issue; others may be more concerned with the running costs associated with power-hungry items. It is also important to consider how to deal with the heat generated by computers. These are issues that companies such as Google consider very carefully, because they use a vast number of computers.

It's interesting to see how power consumption has altered over time as the technologies have changed. A Colossus machine used about 8 kilowatts, whereas the much larger ENIAC required about 170 kilowatts. Just to give you a feel for the relative power levels involved, Table 2.1 lists a few familiar devices and their typical power consumptions.

Table 2.1 Power consumption of some home electrical devices

Device	Power consumption
Electric fire	2 kW
Electric iron	1 kW
46-inch television	120 W
Desktop computer	300 W
Energy-saving light bulb	10 W
Electric kettle	2 kW

Study note 2.2 Abbreviations

In units of power, one kilowatt is 1000 watts. You will notice that in Table 2.1 I have used an abbreviation for the units of power; instead of kilowatts I write kW and instead of watts I write W. In the abbreviation kW, it is worth remembering that the W is upper case and the k lower case.

In order to make a comparison of the power consumptions of these items, you will need to convert all the figures given in kilowatts to watts by multiplying by 1000. For instance, the power consumption of the electric fire is:

$$\begin{aligned} 2\text{ kW} &= 2 \times 1000\text{ W} \\ &= 2000\text{ W} \end{aligned}$$

Activity 2.3 (self-assessment)

(a) Convert the figures given in Table 2.1 for the electric iron and electric kettle from kilowatts to watts.

(b) Which three devices in Table 2.1 use the most power, and what do these devices have in common?

(c) ENIAC required 170 kW of power. How many electric fires could be run for the same amount of power?

(Remember that solutions to the self-assessment activities can be found at the end of this part.)

Activity 2.3 shows that, generally speaking, heating requires a lot of power. Interestingly, in the case of computers, heat is an unwanted by-product – virtually all the power consumed by a computer is converted to heat. Famously, the women who worked at Bletchley Park during the Second World War used the heat from the computers to dry clothes while they were on night duty.

Study note 2.3 Ratio

When I do a calculation such as the one in Activity 2.3(c) above, it always involves dividing one value by the other, which means finding the proportion of one value to another. Such a proportion can be expressed instead as a ratio. So I could have asked 'What is the ratio of the power consumption of ENIAC to that of an electric fire?'. In that case I would write the answer as the ratio 85:1, which is read as 'eighty-five to one'.

Visions of the future

As first-generation computers developed, more people came into contact with them and some began to wonder about the direction in which computers were leading the world. With hindsight, some of the predictions made seem rather pessimistic. Take these two, for example:

> I think there is a world market for maybe five computers.
>
> *(Attributed to Thomas Watson, Chairman of IBM, 1943)*

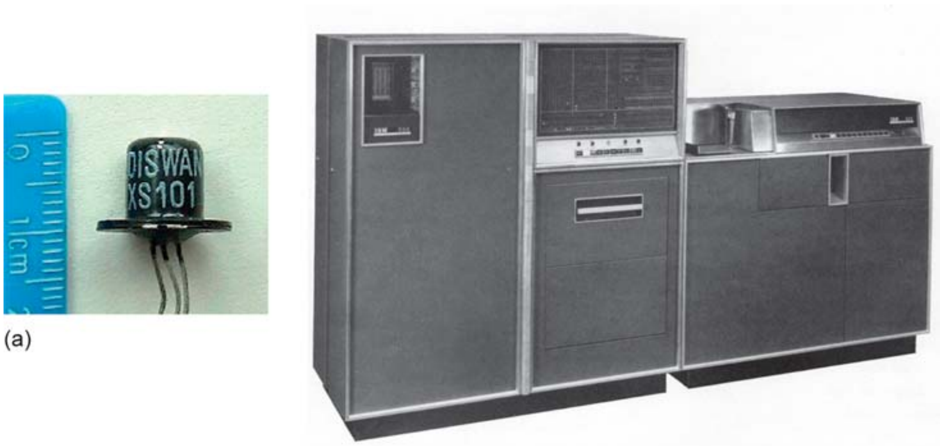
> Where a calculator on the ENIAC is equipped with 18 000 vacuum tubes and weighs 30 tons, computers in the future may have only 1000 vacuum tubes and perhaps weigh 1.5 tons.
>
> *(*Popular Mechanics*, March 1949)*

Throughout much of the first generation, computers were seen as rather mysterious entities. Some people anticipated that they might develop the potential to think for themselves and even emulate human beings. As early as 1950, Alan Turing proposed that a computer should be deemed 'able to think' if it passed what became known as the **Turing test**. This asks whether a computer can hold a dialogue convincingly enough that the human involved in the conversation is fooled into thinking he or she is talking to another human. As yet, no computer is universally accepted as having passed the Turing test.

2.1.2 The second generation

The first generation of computers was defined by the use of valve technology and the assumption that the main purpose of computers was to carry out calculations for code-breaking or other military purposes. During the late 1950s, a second generation began to emerge that used **transistors**.
A transistor is a tiny crystal of silicon (Figure 2.5(a)) that, when processed in various ways, can be made to conduct electrical current and act as a switch by blocking or allowing electrical current, just like a valve. At only a few millimetres in diameter, a transistor occupied less than a thousandth of the

volume of a valve and offered far greater reliability at a fraction of the cost. Additionally, the power consumption of transistors was much lower than that of valves and their switching speed was higher.



(a)

(b)

Figure 2.5 (a) A transistor and (b) a typical transistor computer, the IBM 608

However, second-generation computers were still very expensive both to buy and to maintain – for instance, in 1955 IBM's 608 computer (Figure 2.5(b)) could be purchased for the bargain price of US$83 000 (or about a quarter of that in pounds), much more than the cost of the average house at the time. They were also far from small; in fact, the largest type was referred to as a **mainframe**, after the large cabinets that housed them. Punched cards, such as the one shown in Figure 2.6, were used to program mainframes and to contain data. Although the power consumption of the average computer fell to just a few hundred watts with the emergence of transistor technology, as time went by the complexity of mainframe machines grew and with it their power consumption.

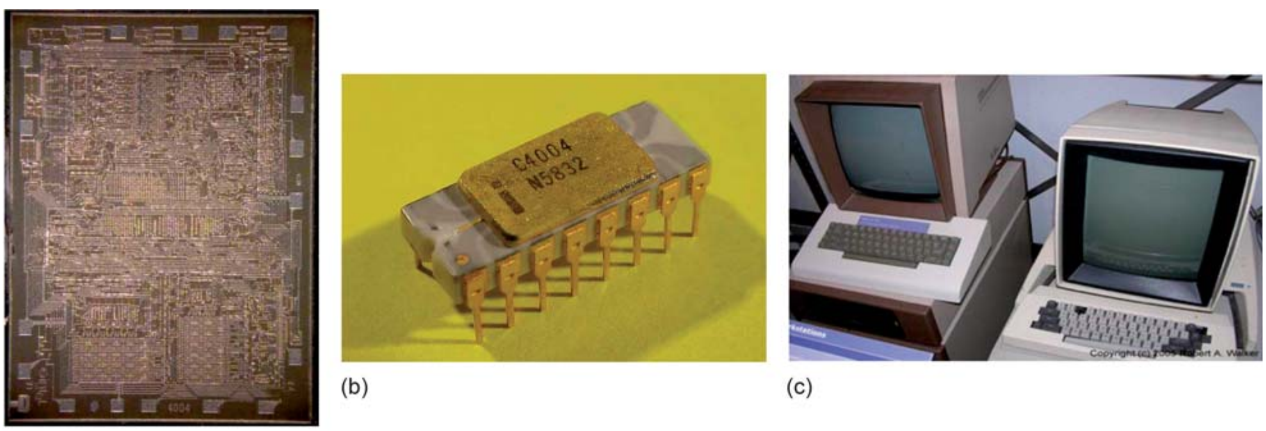
Figure 2.6 A punched card machine with a punchcard, used to program mainframe computers

During the early 1960s a new, smaller type of transistor-based computer, usually referred to as a **minicomputer**, entered the market. A minicomputer came in a case about the size of a wardrobe, and had a number of monitors (display screens) and keyboards connected to it.

Minicomputers were much cheaper to buy and to run than mainframe computers, costing about US$16 000 in 1964. Although they were still far too expensive and took up far too much space to be suitable for personal ownership, they fell within the reach of smaller businesses. Thus the minicomputer expanded the number of computer users considerably and is often seen as marking the beginning of the third generation.

2.1.3 The third generation

The third generation of computers was defined by the emergence of the **integrated circuit (IC)**, otherwise known as the *chip* or *silicon chip*. The first computers that used this technology appeared in 1964. Instead of the individual components (such as transistors) being made from separate crystals of silicon, several components were placed (integrated) on the same crystal. This gave a huge improvement in speed and size, whilst reducing manufacturing costs. Figure 2.7(a) shows the die used to form the Intel 4004 chip, which was the first single-chip microprocessor, and Figure 2.7(b) shows the chip (not actual sizes). Both minicomputers and mainframes moved over to this third-generation technology; Figure 2.7(c) shows a typical computer from this period.



(a) (b) (c)

Figure 2.7 (a) The die that made the Intel D4004 chips (a variant of the 4004); (b) the Intel D4004 chip; and (c) a typical IC-based computer

It was at this point that the way people interacted with computers came close to what we would recognise today. Instead of punched cards for inputting information and paper printouts for output, keyboards began to be used more widely for input and monitors for output, typically with several monitors and keyboards being linked to the same machine. **Terminals**, each of which consisted of a monitor and a keyboard, could be distributed across the country and connected to a single shared computer by telephone lines. These terminals could be used to enter data by filling in forms on screen. This period also saw another significant development in the emergence of the first

floppy disks for storage, making programs and data truly portable for the first time.

Activity 2.4 (exploratory)

Before we move on to more recent developments, I want you to consider how computers came to be widely used in business. For this activity you will need to listen to the audio resource on the module website: extracts from ‘Leo Computer’ by Aleks Krotoski (BBC Radio 4). This is a case study of the development of the LEO computer by Lyons bakery. John Simmons, a senior manager at Lyons & Co, recognised that the computer had the potential to improve business efficiency. Up until then, computers had been developed for use by the military and for scientific purposes. LEO was used for the payroll and for stock control. The latter was particularly important because the goods being stocked and sold were perishable.

The audio recordings last only a few minutes, but you may need to stop and rewind from time to time to ensure that you have understood everything. As you listen to the material jot down some notes about the following questions:

(a) John Simmons considered that people from two particular backgrounds would make good programmers. What were these backgrounds? Does this surprise you at all?

(b) What were John Simmons’ major legacies?

Comment

(a) In those days there were no programmers, so Simmons considered the transferrable skills that could be used in programming. Therefore, he selected people who had skills in either languages or in logic for further training. In doing so, he inadvertently made Lyons an equal opportunities employer in terms of programming.

I was a little surprised at the diverse roles involved in working with LEO even in those early days. People with expertise in linguistics and logic were trained up as programmers, but there were also roles for engineers and data entry jobs.

(b) Simmons' major legacy is the application of computers to business processes. His other important legacy is the holistic view that users and the way they work are central to the use of technology in these processes.

2.1.4 Conclusion

In this section, I have outlined some of the main stages in the development of the first three generations of computers: how the physical forms have changed, the technologies that drove the development of these computers and the different tasks for which they were built. In the next section, I will show you how the development of the microprocessor and other factors led to the popularisation of the personal computer as we know it today.

The fourth generation – computers as we know them

2.2

All the computers I described in the previous section are a long way from today's **personal computers (PCs)**, which trace their origins back to the late 1970s and the emergence of the **microprocessor**, heralding the fourth generation of computers. More recently, the convergence of the microprocessor with mobile technologies has resulted in the 'mobile' computers of today, iPads and tablets. In this section I'll outline the development of the microprocessor and discuss how personal computers have become so prevalent in the world today.

2.2.1 The microprocessor

A microprocessor is essentially a single integrated circuit that contains the **central processing unit (CPU)** of a computer. The CPU can be thought of as the command centre of the computer; amongst other functions, it interprets each program's instructions and organises the storage and retrieval of the data involved. Third-generation computers had the equivalent of a CPU, of course, but at that time the different components of the CPU were usually distributed across several circuit boards (containing the integrated circuits described earlier) and even across several cabinets or equipment racks.

Developing the CPU into a single component brought considerable benefits, including reduced manufacturing and running costs and the potential to manufacture in very large numbers. There were also benefits in terms of processing speed and, later, being able to eliminate some electronic components altogether. For example, when a key is pressed on a keyboard, it is possible for it to seem like a rapid series of presses, because the electrical connection is not always made cleanly; and so early computers required electronic circuits to 'debounce' the keyboard. However, after the development of the microprocessor, it became practical to use software to eliminate this stuttering, and the circuits previously required to do the 'debouncing' could be removed.

With the introduction of microprocessors, personal ownership of a computer became possible for the first time in history. These early personal computers were often referred to as **microcomputers** or just *micros*.

Activity 2.5 (self-assessment)

(a) I have described four generations of computer so far. Identify the technology associated with each generation.

(b) Identify the key benefits that resulted from moving from each generation to the next.

In my answer to Activity 2.5(b) I listed reduction in manufacturing costs as one benefit of the evolution of computers. However, it is important to note that each shift in technology required massive initial investment. The process used for manufacturing microprocessors, for example, has a very high cost associated with the initial design of the components and building the manufacturing equipment, but after this up-front investment the ongoing manufacturing costs are very low. This led to the famous Silicon Valley saying:

> The first chip costs a million dollars; the second one costs a nickel.

The integration of several components into a single component is a common theme in the evolution of computers. The result of this sort of integration is a huge reduction in cost and greatly increased performance. However, this is often at a slight cost in flexibility in that computer manufacturers can no longer access individual components that have been integrated. This means that only the available ‘off the shelf’ configurations can be used. So unless there is high demand for a particular configuration, it’s unlikely that it will be available.

In the late 1970s and early 1980s, a host of popular personal computers emerged that were designed for home use and aimed at mass markets, all based around a narrow range of microprocessors. Typical prices for these machines ranged from about £100 to £500. Many of the companies involved were short-lived; indeed, most never saw the end of the 1980s. One such computer was the BBC Model B (Figure 2.8), which cost around £350. This was part of a series of computers known as the BBC Micros, which were built by the Acorn Company in a joint enterprise with the BBC – known as the BBC Computer Literacy Project – that aimed to bring computers into schools with accompanying television programmes.

However, it was the IBM 5150 desktop machine and those that followed which became synonymous with the ‘personal computer’ or PC as it became the standard for business computers. Only Apple’s Macintosh machines retained a sizeable share of the computing market in that they were popular in education and desktop publishing.

Figure 2.8 A BBC Model B Micro

2.2.2 Into the 1990s: strands of computing

By the early 1980s, several different kinds of computer were emerging, and these continued to develop into the 1990s. There isn't space here for me to describe them all in detail, but in very broad terms you can think of them as making up four strands of evolution, differing in size, cost and purpose. These four strands were as follows.

- Small personal computers, designed primarily for home use and small businesses. Aside from games (which were increasingly attracting people to computers), these were used for **applications** such as word processing, spreadsheets, databases and the like.
- Larger, more expensive **workstations**, which were used for modelling, running sophisticated graphics applications and other similar processor-intensive tasks. These machines were often interconnected in order to share processing power, and so each was a **multi-user computer**, meaning that many users could simultaneously access its resources (CPU or memory, for example).
- Minicomputers, used for many of the same tasks as workstations. One central minicomputer had distributed terminals, allowing numerous users to interact with the computer. By the early 1990s, there was considerable overlap between the roles of workstations and minicomputers. For example, many universities replaced their minicomputers with networks of workstations, to provide email and administrative services alongside their existing role in supporting academic activities such as mathematical modelling.
- The highly expensive mainframes and **supercomputers** used by the defence sector, academics, very large businesses and government organisations. Supercomputers gained prominence in the 1990s; they were typically used to do complex and numerically intensive work such as modelling weather systems.

Figure 2.9 shows the Sunway TaihuLight, the fastest computer in the world at the time of writing (2016). In the meantime, mainframes were increasingly being used for tasks relating to large databases, such as payroll management.

Figure 2.9 The Sunway TaihuLight supercomputer located in the Chinese National Supercomputing Center in the city of Wuxi, in Jiangsu province, China

The components of a personal computer

The CPU in the form of a microchip remains the key hardware component of computers today. Although more and more elements of the computer have found their way onto the CPU over time, the fundamental layout of most personal computers has remained unchanged since the time of the micro in the late 1970s. Figure 2.10 shows some of the basic components of a desktop personal computer.

You can see that the components 'inside the box' include:

- the CPU
- the **main memory**, sometimes referred to as **random-access memory (RAM)**, which provides temporary storage for data and program instructions whilst a program is running
- the motherboard, on which are located the CPU and main memory as well as other key components – the motherboard also provides all the connections for the **peripheral devices**
- the **hard disk drive**, which provides persistent storage for data and programs (i.e. it stores data until you choose to delete it)
- a CD/DVD drive (on some computers), which reads from and writes to CDs/DVDs.

Then there are external peripheral devices, such as the keyboard, printer and monitor, which are connected to the computer by cables. Also, devices such

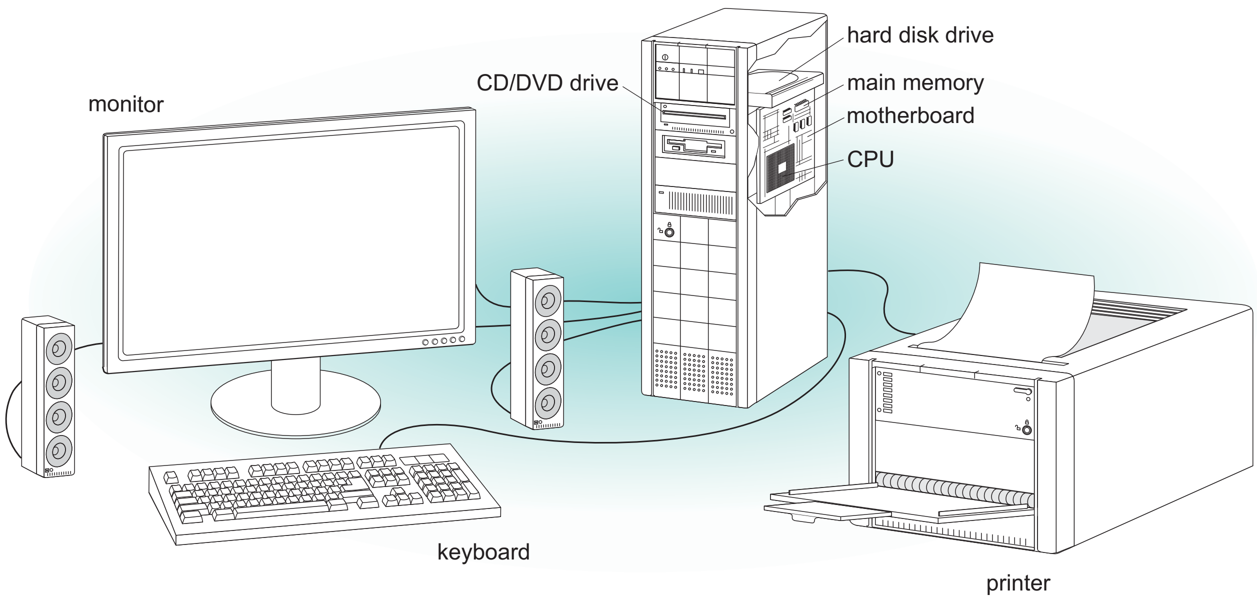


Figure 2.10 The basic components of a desktop personal computer

as USB (Universal Serial Bus) memory sticks and external hard drives are used to store data outside the computer, such as documents and photographs.

Study note 2.4 Backing up your work

You could lose everything stored on your computer to hard disk failure or the theft of a laptop. So make sure you back up your files regularly.

One way is to save your files to an external disk. This strategy can be hard work as you should in theory have a minimum of two backup disks that you use in rotation, with the unused disk being stored in a different building from the current disk. If you do what most people do – have just one backup disk that sits next to the computer – you're exposed to losing both the original and the backup to fire or theft.

The easiest way to back up is to use a cloud-based storage system such as Dropbox or OneDrive for all your important files. These keep copies of your documents both on your computer and on an internet server, with further sets on other computers you use. They synchronise the sets of documents so that actions to add, modify, move or delete files are replicated in all the sets.

Activity 2.6 (exploratory)

OpenStudio is an online area that allows you to share content with other students on your module; it can be shared with your tutor group, or used as an individual work portfolio. You will be uploading an image to OpenStudio and posting a comment. You will find the details, along with a link to OpenStudio, on the module website.

2.2.3 The proliferation of the personal computer

You saw in Part 1 that personal computers have come to play a hugely important role in society. I have already presented a brief history of personal computers, from their emergence in the 1970s to the more modern machines of the 1990s. These early personal computers were the ancestors of those we are familiar with today; developments in technology led to reductions in their cost and size, putting them within the reach of more and more people. But why did more and more people actually want to use computers? Aside from technology, just what propelled the development of the personal computers that many of us use every day? I will discuss three key factors: killer apps, increasing usability, and the rise of the internet.

Killer apps

A **killer app** is any application considered so desirable that people would purchase a computer purely to use it. One of the first killer apps was the spreadsheet application VisiCalc, which simulated a physical spreadsheet by capturing, displaying and manipulating data arranged in rows and columns. VisiCalc was released in 1979 for the Apple II computer and had a massive impact on sales of the Apple II – not surprisingly, sales to the financial sector were colossal. A similar thing happened with the launch of Lotus 1-2-3 (another spreadsheet application) for the IBM PC. Software became the key to marketing personal computers.

Interestingly, though, exactly how software would come to dominate and steer the market was not widely understood in the 1980s. When IBM launched the 5150 personal computer in 1981, they gave it what was described as *open architecture*; this meant that IBM made public various elements of the computer's design. The aim was to enable other companies to produce software and peripheral devices, and thus to ensure that there were plenty of useful things the machine could do. Yet this open architecture approach eventually led to two things that IBM didn't foresee:

1 It enabled other companies to produce clone machines at a fraction of the price that were able to run the same software as their IBM counterparts.
2 It enabled Microsoft, who produced the operating system for the IBM machine, to dominate the software market, because this operating system was used on all the clone machines as well.

With so many manufacturers all producing compatible machines, the software market for games and office applications boomed. To this day, there remain two dominant types of personal computer: Microsoft Windows-based machines, which are derivatives of the IBM clones, and Apple machines, which retain a smaller share of the market.

Increasing usability

All computers have an operating system, which can be thought of as mediating between the user and the electronics of the machine. The operating system manages the computer's different hardware components and peripheral devices, provides a means for the user to interact with the computer, and organises the running of application software (see Figure 2.11). You will probably be familiar with your computer's operating system – perhaps Microsoft's Windows or Apple's macOS. Because it facilitates your interaction with your computer, the operating system makes a crucial contribution to your computer's *usability* – put simply, how easy it is to use.

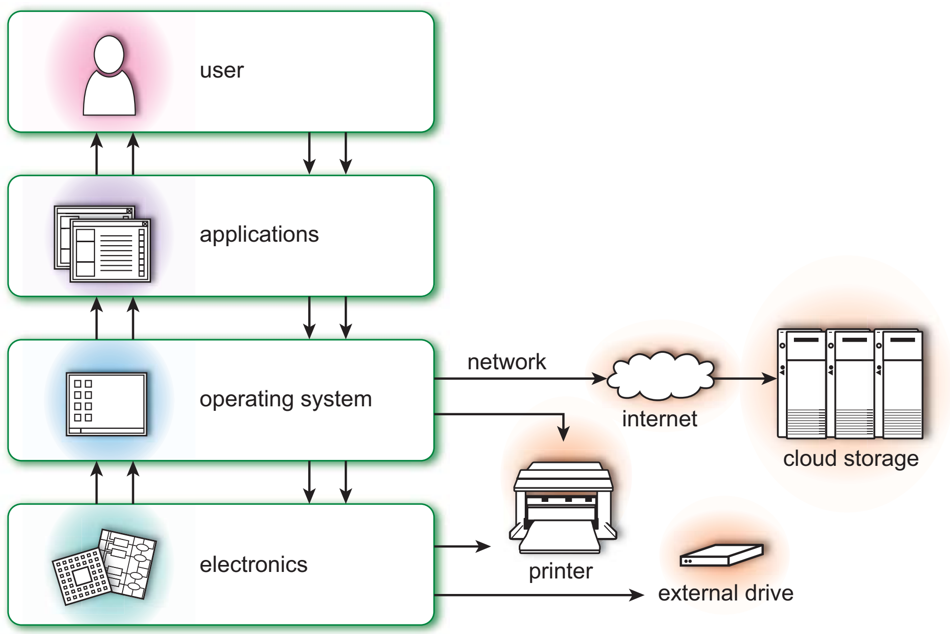


Figure 2.11 The operating system

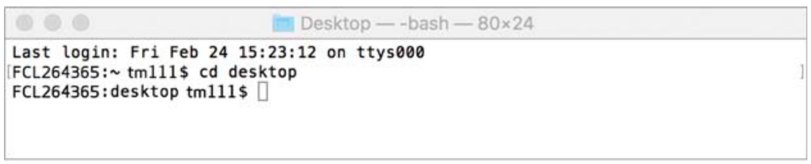
Activity 2.7 (exploratory)

Identify your computer's operating system (and version number if it has one). If you are not sure, watch your screen next time you start up your computer; the operating system information is usually one of the first things to appear. If you are using a Mac, you can find this information in 'About this Mac' in the Apple menu.

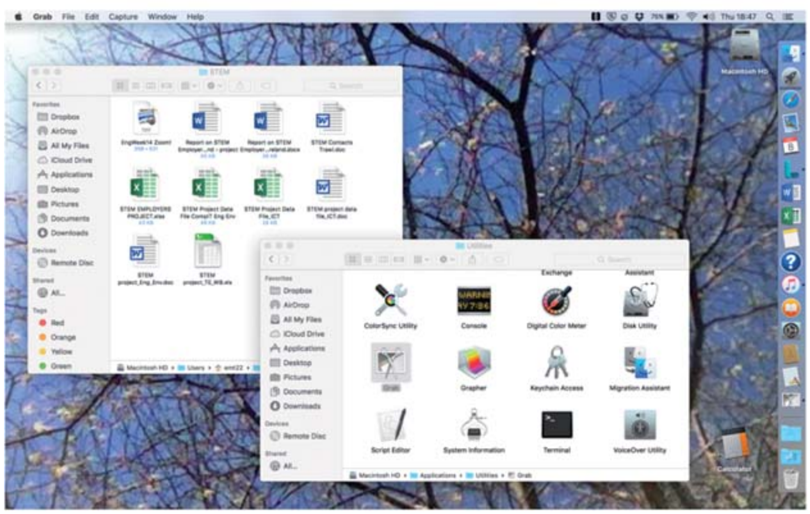
Comment

I am writing this material on a Mac computer running the OS X operating system version 10.11.6.

Over the years, operating systems have grown increasingly sophisticated. Their key development in terms of usability was probably the **graphical user interface (GUI)**, which was first seen on the Apple Lisa machine of the early 1980s. The GUI enabled users to interact with graphics on the computer screen by pointing and clicking using a mouse. Before that, it had been necessary to remember a series of command codes, which had to be typed into a text-based *command-line interface*, in order to do anything. Thus the GUI represented a huge step forward in usability. Figure 2.12 shows the difference in appearance between a command-line interface and a GUI.



(a)



(b)

Figure 2.12 (a) A command-line interface; (b) a GUI

The rise of the internet

The internet has its origins in a 1969 research project known as the Advanced Research Project Agency network or ARPANET, which you will learn more about in a later part. However, the internet as we know it emerged in the 1980s and grew in public popularity with the birth of the web in 1992. The personal computer industry was driven in a new direction, as connectedness gradually became essential. Today, most personal computers rely on being connected to the internet to maintain their integrity, using it to update virus scanners or the operating system. This assumption of connectedness, as you will see later in this part when I discuss cloud computing and mobile devices, is leading to a very different future for the personal computer.

2.2.4 Conclusion

In this section I have outlined the main features of the fourth generation of computers and shown how the personal computer gradually became a standard feature of the workplace and the home. I hope you can see that the history of computers involves ever increasing complexity as more and more manufacturers came into the market. I'll say a little more about where Computing and IT developments are heading in Section 2.5, but in the next section I'll talk about the phenomenon known as exponential growth and its relevance to the development of computers.

If you would like to know more about the development of computers, you can visit the National Museum of Computing at Bletchley Park. Their website is also very interesting and informative.

2.3 Smaller, faster, cheaper

In this section, I'm going to introduce you to an amazing prediction about the rate of development of computer components, made as early as 1965. This will require you to learn a little about the numerical concept of *exponents*. I will then indicate the wider relevance of exponents to computing by explaining how they are used when representing computer speeds.

2.3.1 Moore's law

Computers today are smaller, faster and cheaper than they have ever been. As you learned in the last section, I could have made this statement at pretty much any time over the last 30 years and it would have been true; it's doubtful that any other product has maintained such rapid development over such a long period of time. But just how fast has computer development been? In 1965, Gordon Moore, the founder of the giant Intel Corporation, wrote in *Electronics* magazine that he expected the density of electronic components in an integrated circuit on a silicon chip to double every year for at least the next ten years.

Activity 2.8 (self-assessment)

(a) In which generation of computers did Moore make his prediction?

(b) What technological development led this generation into the next?

Whether we look at the CPU or at memory, the fundamental component of the computer is the transistor, and it has been since the emergence of the second generation. The incorporation of many transistors onto a single silicon chip, in the form of an integrated circuit, started a process of miniaturisation that continues to this day. More and more transistors are placed in a given space; as the distance between the transistors shrinks, so the speed of communication between them increases and the cost per transistor falls.

When he made his prediction, Moore felt that he could see how the technologies needed to achieve it would evolve for the next five to ten years. In fact, the prediction was slightly out, as the density of components has doubled every two years rather than the year that was the rate in 1965, and it is this rate of growth that is now associated with Moore.

Nevertheless, it is quite astonishing that development has continued at this rate for 50 years after the initial prediction. In fact, behind this lies a massive amount of investment to ensure that the trend continues. At some point, **Moore's law**, as it is called, moved from being a prediction to being the target for an industry. Intel's Core M processor, released in 2015, holds 1.3 billion transistors that each measure about 14 nanometers (nm) in size. To give you an idea of the scale: a flu virus is 20 nm in diameter.

The graphs in Figure 2.13 show two different representations of Moore's law. In both cases, the horizontal **axis** shows elapsed time and the vertical axis the number of components (how many transistors will fit on a given area of silicon). However, in graph (b) the vertical axis is calibrated differently; that is, its scale is different.

Activity 2.9 (exploratory)

Look at the two vertical axes of the graphs in Figure 2.13 and describe the difference in your own words. Why is graph (b) a more useful representation than graph (a)?

Comment

Here's my attempt. The difference is that on the vertical axis of graph (a) the increments are in a linear sequence (10 000, 20 000, 30 000, etc.), but for graph (b) the increments increase in multiples of ten (10, 100, 1000, etc.).

Graph (a) is useful in that it gives a visual indication of the growth rate, but it's very hard to read off the small values before the year 2000. Graph (b) is easier to read and can be used to predict future values; we call this type of prediction *extrapolation*.

A scale on which the increments increase in multiples of a fixed amount (ten in this case) is referred to as a **logarithmic scale**, and any graph that has such a scale along *one* of its axes is described as 'semi-logarithmic'. Graphs of this type are very widely used to represent data that varies over a large range. When you look at a graph, it's well worth checking whether any of the scales are logarithmic – it is easy to overlook this and so misunderstand the way the data is varying.

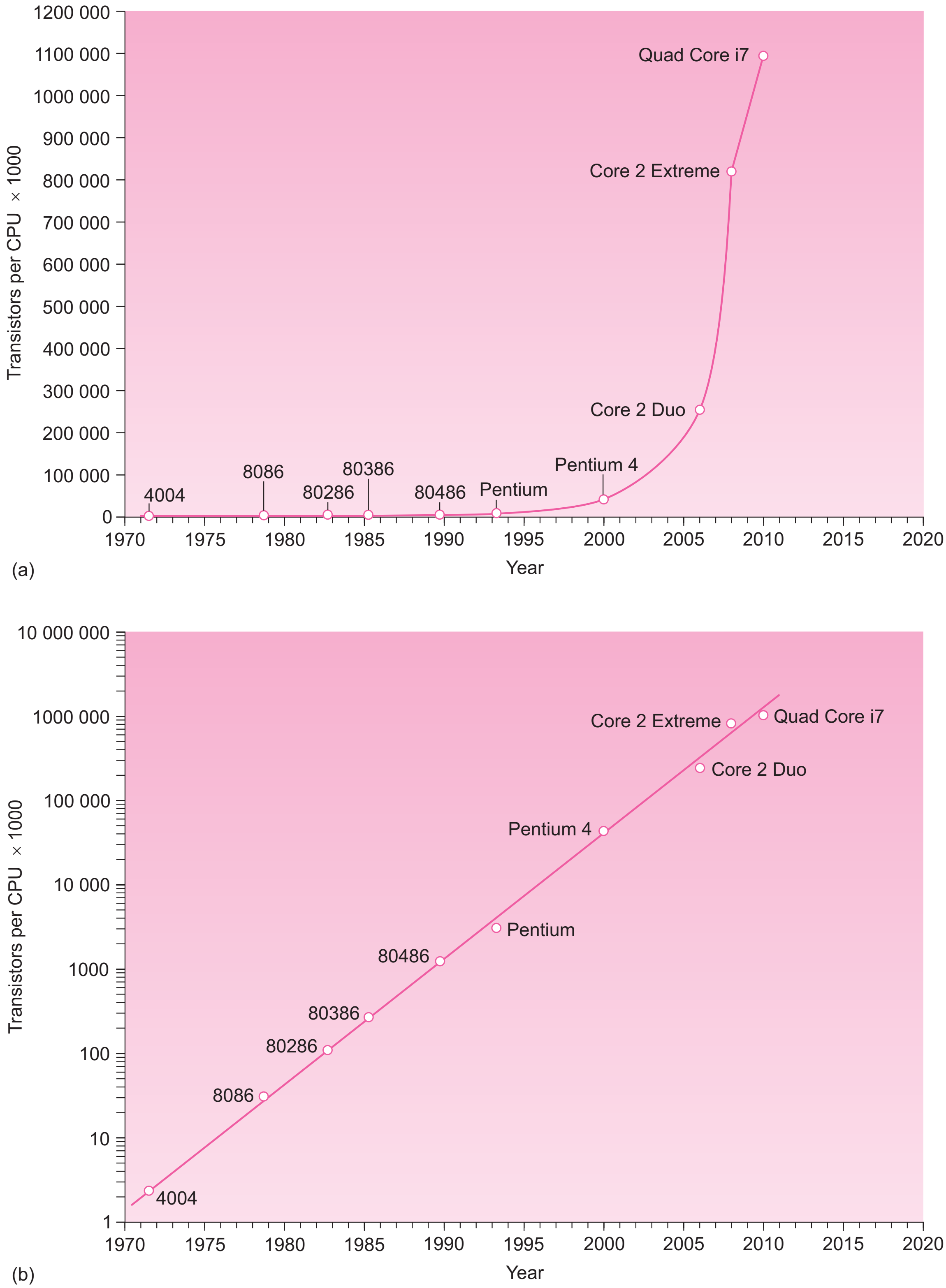


Figure 2.13 Two different representations of Moore's law

The legend of the Ambalapuzha Paal Payasam

To help you appreciate the significance of the doubling in performance every two years suggested by Moore's law, I'd like you to think about the ancient legend of the Ambalapuzha Paal Payasam. In this legend, the god Krishna (in disguise of course!) plays chess with a king. The terms of the wager are that the loser should pay in rice, by placing one grain of rice to correspond to the first square of the chessboard, two grains to the second, four to the third and so on – each time doubling the number of grains of rice placed on the previous square (see Figure 2.14). Inevitably, the king agrees to the wager and loses.

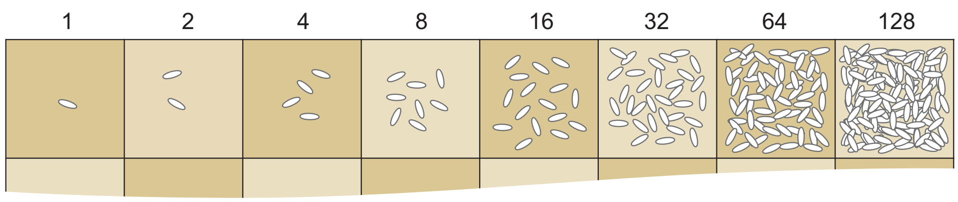


Figure 2.14 Doubling the number of grains in each square

There are 64 squares on a chessboard. By the time the king is placing rice corresponding to the 21st square, he requires over a million grains (this is 2 multiplied by itself 20 times). At the 64th square, it is almost meaningless to talk in terms of grains of rice, but the total amount required for the board weighs over 400 billion tonnes.

Activity 2.10 (self-assessment)

Explain why the number of grains of rice on the 21st square is 2 multiplied by itself 20 times.

This legend goes some way towards showing how remarkable it is that the computer industry doubles the number of components in a given area every two years. And it doesn't stop there. Alongside the development of these components come a host of other developments in communications and magnetic storage media that have at times matched, and even occasionally outstripped, the integrated circuits to which Gordon Moore was referring.

Exponential growth

Moore's law and the legend of the Ambalapuzha Paal Payasam both illustrate **exponential growth**. In both cases, the result at each step is the previous result multiplied by 2. So, for Moore's law we start with the area occupied by one component, then multiply the number of components by 2 for each two-year period. Thus I have:

1 component initially

$1 \times 2 = 2$ components in that area after 2 years

$2 \times 2 = 4$ components in that area after 4 years

$2 \times 2 \times 2 = 8$ components in that area after 6 years

$2 \times 2 \times 2 \times 2 = 16$ components in that area after 8 years, and so on.

A more efficient way of writing the numbers above is as

2^1

2^2

2^3

2^4, and so on.

That is, we write 2^n to mean '$2 \times 2 \times ... \times 2$' with n 2s, where n is any whole number greater than 0. The superscript number beside the 2 is called the *exponent* or **power**; a number that can be written in the form 2^n is called an *exponent of 2* or *power of 2*. So 2, 4, 8, 16, etc. are all powers of 2, since:

2^1 is just 2

2^2 is 2×2, i.e. $2^2 = 4$

2^3 is $2 \times 2 \times 2$, i.e. $2^3 = 8$

2^4 is $2 \times 2 \times 2 \times 2$, i.e. $2^4 = 16$, and so on.

We say that a quantity increases *exponentially* if it can be described using numbers that involve exponents. You've seen that Moore's law is an example of exponential growth – in this case, after n lots of 2 years, the number of components in an area that originally contained just one component is 2^n.

Study note 2.5 Exponential notation

The use of numbers involving exponents is very useful, and you will come across it occasionally in TM111. Furthermore, its use extends beyond computing; almost any subject that involves very large numbers, including astronomy, chemistry and finance, uses powers of 10 to express the number of zeros.

Activity 2.11 (self-assessment)

Based on the number of components in a given area doubling every 2 years, how many components would there be in an area after 16 years if it originally contained one component? You may wish to use a calculator for this activity.

Of course, exponential growth is not limited to powers of 2. In different cases of exponential growth, powers of other numbers can be involved. So, more generally, we write x^n to mean 'x multiplied by itself n times'.

For example:

$3^2 = 3 \times 3 = 9$

$4^3 = 4 \times 4 \times 4 = 64$

$10^2 = 10 \times 10 = 100$

Activity 2.12 (self-assessment)

Find the value of each of the following: 2^6, 2^9, 10^3, 10^4, 8^3. You may wish to use a calculator for this activity.

Activity 2.13 (self-assessment)

You've seen that $10^2 = 100$, $10^3 = 1000$ and $10^4 = 10\,000$.

What is 10^5?

Can you explain the pattern?

Write 1000 000 as a power of 10.

So, for example, rather than writing one hundred thousand billion or 100 000 000 000 000, I could simply write 10^{14}. This notation also reduces the risk of miscounting zeros. You might know that in 2014 the annual cost of cybercrime globally was estimated to be over US\$400 billion. Since 400 billion is equal to $4 \times 100\,000\,000\,000$, this can be written more simply as 4×10^{11} US dollars.

It also makes sense to talk about powers that are negative, zero or fractional (such as 2^{-3}, 10^0 or $5^{1/4}$). For example, any number (except 0) to the power of 0 is 1. You will learn more about negative powers of 10 and how they can be used to express very small numbers in a later part.

Eventually, exponential growth always has to stop because it encounters some physical limit, be it the availability of grains of rice or of atoms in a silicon chip. The fact that this has not yet happened in the computing industry is remarkable, but how long can it be before we reach a point where it is impossible to make things any smaller? We might encounter such a limit because it is not possible to manufacture components smaller than a certain size – less than an atom wide, say – or because the components wouldn't function reliably below a certain size. On the other hand, people have been saying this for many years about Moore's law, although, currently, the 'law' is believed to be faltering as the benefits of making things smaller have been decreasing while the costs have been rising.

2.3.2 Naming large numbers

Earlier in this part, when you looked at power consumption, you came across the idea that a kilowatt is equal to a thousand watts. The prefix 'kilo' is a common one.

Inevitably, given Moore's law, when talking about computers we need larger numbers than we encounter in the everyday world. The speed of modern computers is difficult to quantify, because the speed of a computer depends on the nature of the task it is performing and because different machines are optimised for different tasks. However, a common approach is to identify the number of instructions that the CPU can carry out in a given interval of time – often 1 second. Very large numbers are involved, usually expressed as powers of 10. For example, you will hear the terms *megaflops* (which shortens to *Mflops*), *gigaflops* (*Gflops*) and *teraflops* (*Tflops*) used in this context.

Table 2.2 shows a list of commonly used prefixes and their abbreviations.

Table 2.2 Prefixes and their abbreviations

Prefix	Abbreviation	Multiply by…	Power of 10
kilo	k	1000	10^3
mega	M	1000 000	10^6
giga	G	1000 000 000	10^9
tera	T	1000 000 000 000	10^{12}
peta	P	1000 000 000 000 000	10^{15}
exa	E	1000 000 000 000 000 000	10^{18}
zetta	Z	1000 000 000 000 000 000 000	10^{21}
yotta	Y	1000 000 000 000 000 000 000 000	10^{24}

The main prefixes in everyday use at present are the first four: kilo, mega, giga and tera. The speed of a typical personal computer might be tens of gigaflops. At the higher end of the scale, supercomputers are capable of reaching speeds measured in petaflops.

Box 2.2 'Flops'

The 'flops' part stands for 'floating-point operations per second' which is a way of handling tremendously long numbers relatively easily in computing. The prefixes 'giga' and 'tera' indicate the power of 10 involved: giga indicates 10^9 (or a billion) and tera indicates 10^{12}.

The prefixes shown in Table 2.2 can be used not just for computer speed but in any other situation involving large numbers – for example, when measuring power (kilowatts, megawatts, etc.). However, in Section 2.6 you will learn that, when applied to computer memory, these prefixes don't have exactly the same meanings as those given in Table 2.2.

In addition to prefixes to cover large scales, there are also prefixes to cover very small scales, which you may already have seen in use. For example, 'milli' (abbreviation m) means 'divide by 1000' and 'micro' (abbreviation µ) means ‘divide by 1 000 000’. You will learn more about these prefixes in a later part.

2.3.3 Conclusion

In this section I introduced you to Moore’s law, which is an example of exponential growth. I then looked at exponential notation more generally, before considering the prefixes that are commonly used to describe large powers of 10. The next section explores binary numbers and how they are stored in a computer.

2.4 The language of computers

Having looked at some of the history and exponential growth of computing power, it's now time to look at what goes on inside a computer. This section introduces you to binary logic and how binary numbers represent information inside computers.

2.4.1 Binary logic

Binary logic underpins most forms of digital technology. It was developed in the mid-nineteenth century by George Boole (Figure 2.15), an Irish mathematician and philosopher, long before anyone considered making a computer using electronic components. At the time, Boole was interested in exploring the fundamental nature of truth, so he developed a system involving symbols – a *symbolic logic* – to help him reason about truth and falsity. This system is often referred to as *Boolean logic* or **binary logic**, and it focuses on the manipulation of 1s and 0s (1 to represent TRUE and 0 to represent FALSE).

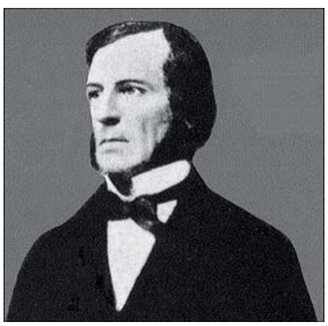


Figure 2.15 George Boole (1815–1864)

When trying to understand the nature of the digital world, it is useful to know something about the binary numbers that Boole based his algebra upon, because binary numbers are the form in which information is held in a computer. As you saw in Section 2.1.1, a *binary number* is a sequence of digits, each of which is either 1 or 0. For example, 100011 can be considered as a binary number. When stored in a computer, each digit of a binary number is referred to as a **bit**, which is a contraction of the words *binary digit*.

There are a number of different ways in which it is possible to represent 1s and 0s in the physical world, the most obvious representation being a switch. In one position (or state), a switch is said to be 'off' (0), and in another 'on' (1) (Figure 2.16).

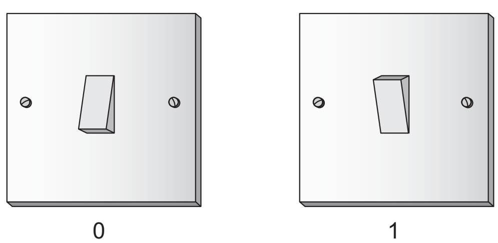


Figure 2.16 Using switches to represent 0s and 1s

Clearly, a single switch isn't going to make the world's most sophisticated logic machine, but what happens if I have more switches? This would allow me to represent several 1s and 0s at the same time – in fact, I can represent any binary number, as long as I have enough switches. The row of switches in Figure 2.17, for example, represents the eight-bit number 00110101. This is the form in which data is stored, manipulated and transferred in computers.

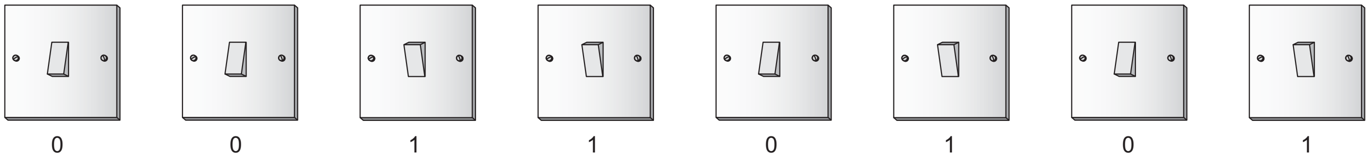


Figure 2.17 A row of switches representing an eight-digit binary number

How many?

You can see from the preceding discussion that the more switches (or bits) I have, the more different binary numbers I can represent. If I use a single bit, then I can represent only two different numbers: a 1 or a 0. If I use two bits, then I can represent four different binary numbers:

00

01

10

11

Activity 2.14 (self-assessment)

How many different binary numbers can be made with three bits? To start you off, 000, 001 and 010 are all possibilities.

There's a pattern here that is very like the pattern you met earlier when I discussed Moore's law. Every time the number of bits is increased by one, the number of possible binary numbers doubles. So:

with 1 bit, I can form 2 binary numbers

with 2 bits, I can form 4 binary numbers, that's 2 × 2 binary numbers

with 3 bits, I can form 8 binary numbers, that's 2 × 2 × 2 binary numbers

with 4 bits, I can form 16 binary numbers, that's 2 × 2 × 2 × 2 binary numbers, and so on.

Just as with Moore's law, this pattern can be represented using powers of 2:

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$, and so on.

Here you can see that the number of possible binary numbers that can be made from n bits, where n is any whole number bigger than 0, is 2^n.

Activity 2.15 (self-assessment)

In computing, a set of eight bits is referred to as a **byte**. How many different binary numbers would I be able to form using eight bits?

2.4.2 Base 10: decimal numbers

In the everyday world, we use a system of counting and arithmetic that relies on having ten different symbols for digits, i.e. 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. When we count up and arrive at 9, the next number is represented by placing a 1 one place to the left and a 0 to the right of it, giving 10.

Here, the 1 represents the number of tens I have; in other words, this can be read as *one* ten plus *zero*, or ten. I can extend this system indefinitely by simply adding columns to the left: hundreds, thousands, ten thousands, hundred thousands and so on. Each column has ten times the value of the column to its right. It can help to visualise the columns set out as follows. (I mentioned the fact that $10^0 = 1$ in Section 2.3.1.)

Ten thousands	Thousands	Hundreds	Tens	Units
10 000	1000	100	10	1
$= 10 \times 10 \times 10 \times 10$	$= 10 \times 10 \times 10$	$= 10 \times 10$		
$= 10^4$	$= 10^3$	$= 10^2$	$= 10^1$	$= 10^0$

As I've said, each column value (or *weighting*, as I will call it) in this table is found by multiplying the weighting to the right by 10. The first weighting (on the far right-hand side) is 1. The fact that in this system ten digits (0 to 9) are used, with each column value being found by multiplying the weighting to the right by 10, leads to it being called a *base 10 numbering system* or a **decimal numbering system**.

Numbering systems that work in this way, using a set of digits or symbols and weighted columns, are called *place value systems*. Not all numbering systems are place value – Roman numerals, for instance. Systems such as the Roman system make arithmetic very difficult indeed; try adding XVIII to MCVI if you want to see what I mean. But that's another story, and one that is well outside the scope of this module!

2.4.3 Base 2: binary numbers again

You've seen that by using any whole number as a base, it is possible to create a system whose numbers are equivalent to those in the decimal system we use every day. Using 2 as the base has particular relevance to computers.

The numbers of the base 2 numbering system are strings of the digits 0 and 1: for example, 11001011. This should look familiar! Base 2 numbers are the binary numbers you encountered earlier.

You have seen how computers can store binary numbers. Now you can also begin to see how computers use binary numbers to represent the decimal numbers we use in everyday life.

For the binary number system, the table of columns begins as follows.

Eights	Fours	Twos	Units
8	4	2	1
$= 2 \times 2 \times 2$	$= 2 \times 2$		
$= 2^3$	$= 2^2$	$= 2^1$	$= 2^0$

Activity 2.16 (self-assessment)

What would be the values in the next three columns to the left?

Binary to decimal

A table such as the one above can be used to convert binary numbers to their decimal equivalents. Below, for example, I've positioned two binary numbers, 1001_2 and 111_2, in the table.

Note the use of the subscript 2 to indicate a binary number. You should omit the subscript only if the context makes clear which number base you are working in.

Eights, 8	Fours, 4	Twos, 2	Units, 1
1	0	0	1
0	1	1	1

For each binary number, I multiply each digit in each column by the weighting at the head of that column to yield a decimal number. Adding these decimal numbers together produces a decimal number equivalent to the binary number I started with. So for 1001_2, I have:

$1 \times 8 = 8$

$0 \times 4 = 0$

$0 \times 2 = 0$

$1 \times 1 = 1$

Adding these numbers together gives $8 + 0 + 0 + 1 = 9$, so $1001_2 = 9_{10}$.

In other words, I've found out that the binary number 1001 is equivalent to the decimal number 9.

Activity 2.17 (self-assessment)

(a) Try this procedure with the second number in the table, 111_2.

(b) Convert the following binary numbers to their decimal equivalents.

(i) 10_2

(ii) 10001_2

(iii) 11000_2

(iv) 11111_2

There is potential for confusion when speaking about numbers with different bases. The number 10_2 should never be referred to as ten. It is best to say either 'one zero' or 'binary two'.

> There are 10 types of people in the world: those who understand binary numbers and those who don't.
>
> *(Anon)*

Converting decimal to binary

So far, you've learned how to convert a binary number into its decimal equivalent. But, of course, to represent decimal numbers in binary form on a computer requires a conversion the other way round. That is, it's necessary to find the binary equivalent of a decimal number.

In this example, I am going to show you how to convert the number 157 to binary. There is a procedure for converting from decimal to binary that relates to the tables you have seen above. First I write down the weightings for each column in the binary number system, from right to left, up to the largest value that does not exceed the decimal number I am converting. So, in this example where I am converting 157 to binary, I start by writing the following.

128	64	32	16	8	4	2	1

(The numbers in these columns are increasing powers of 2, from right to left.)

Activity 2.18 (self-assessment)

What would be the number in the next column to the left? Why do I stop at 128?

Now I subtract the weighting 128 from 157 and place a 1 in the 128 column. This gives a remainder of 29. The table expresses the fact that
$157 = 1 \times 128 + 29$.

128	64	32	16	8	4	2	1
157 − 128 = 29							
1							

Now I think about the remainder, 29. I move along the columns to the right until I find the first weighting that can be subtracted from 29 without leaving a negative remainder. Since 64 is too large to subtract from 29, I place a 0 in the 64 column. Likewise, 32 is too large to subtract from 29, so I also place a 0 in the 32 column. I can subtract 16 from 29, with remainder 13, so I place a 1 in the 16 column. The table now expresses the fact that
$157 = 1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 13$.

128	64	32	16	8	4	2	1
157 − 128 = 29			29 − 16 = 13				
1	0	0	1				

The remainder is 13. Again, I move along the columns to the right. I can subtract 8 from 13, with remainder 5, so I place a 1 in the 8 column. The table expresses the fact that
$157 = 1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 5$.

128	64	32	16	8	4	2	1
157 − 128 = 29			29 − 16 = 13	13 − 8 = 5			
1	0	0	1	1			

The remainder is 5. I can subtract 4 from 5, with remainder 1, so I place a 1 in the 4 column. The table expresses the fact that
$157 = 1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 1$.

128	64	32	16	8	4	2	1
157 − 128 = 29			29 − 16 = 13	13 − 8 = 5	5 − 4 = 1		
1	0	0	1	1	1		

The remainder is 1. I can't subtract 2 from 1, so I place a 0 in the 2 column. I can subtract 1 from 1, with remainder 0, so I place a 1 in the 1 column – and I am finished. In its final form, the table expresses the fact that
$157 = 1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$.

128	64	32	16	8	4	2	1
157 − 128 = 29			29 − 16 = 13	13 − 8 = 5	5 − 4 = 1		1 − 1 = 0
1	0	0	1	1	1	0	1

I can now read off the column entries to find that the binary equivalent of the decimal number 157 is 10011101. Large binary numbers are often written in groups of four digits to help with readability. Although the term is rarely used, a set of four bits is called a *nibble*. So $157_{10} = 1001\ 1101_2$.

Note that you could check the answer here by starting from the binary number $1001\ 1101_2$ and converting it – using the technique you practised earlier – to its decimal equivalent. You should arrive back at the decimal number 157.

When converting from decimal to binary, you don't have to give tables at each intermediate step, as I've done above. You can simply show your working (the remainders) within one table. Try this in the following activity.

Activity 2.19 (self-assessment)

In the table below I've started the conversion process for the decimal number 178. Complete the table and write down the equivalent binary number.

128	64	32	16	8	4	2	1
178 − 128 = 50							
1							

Activity 2.20 (self-assessment)

Convert the following decimal numbers to binary.

(a) 199

(b) 64

(c) 328

(d) 1000

2.4.4 Measuring data storage – bits and bytes

When I discussed prefixes in Section 2.3.2, the examples I chose were of computer speed. You learned, for example, that a megaflop represents a speed of 10^6 – a million – floating-point operations per second.

Computers store data in sets of eight bits which, as you saw in Activity 2.15, are called bytes. Following the speed examples, you might reasonably think that this means, for example, a 3 MB (megabyte) memory stick can store 3×10^6 or 3 000 000 bytes in its memory. However, the computing and telecommunication industries don't use terms such as kilo and mega consistently.

Activity 2.21 (exploratory)

If you have a calculator to hand, calculate the values of 2^{10} and 2^{20}.

(Hint: use the x^y button on a scientific calculator to enter the power, e.g. '2 x^y 10' to find the value of 2^{10}.)

Comment

$2^{10} = 1024$ and $2^{20} = 1048\,576$.

I think you'll agree that these values are temptingly close to 1000 and 1000 000, or 10^3 and 10^6.

When talking about storing data, the industry often uses the term kilobyte (KB) to mean 1024 bytes, the term megabyte (MB) to mean 1048 576 bytes, and so on. Note the capital K denoting 2^{10}, as opposed to the lower-case k that stands for 10^3.

I have a memory stick which can store 3 megabytes (MB) of data. I can calculate the number of bits in 3 MB by:

- firstly, converting 3 MB into bytes which is 3×2^{20} or $3 \times 1048\,576 = 3145\,728$ bytes
- then converting the bytes into bits, 3145 728 bytes $\times$ 8 = 25 165 824 bits.

Therefore, 3 megabytes = 25 165 824 bits.

Activity 2.22 (self-assessment)

(a) How many bytes can be stored by a device that has a memory capacity of 512 megabytes (MB)?

(b) How many bits of data is this?

You will probably need to use a calculator for this activity.

Oddly, for data *transmission,* as opposed to *storage,* the industry uses 1000 and 1000 000 for kilo and mega, respectively – and, just to add a further complication, gives quantities in bits rather than bytes. For example, your broadband could be described as a 28 mbps (megabits per second) connection for data transmission – and this is 28 000 000 bits per second. It is important to recognise the difference between the units used for data *storage* and data *transmission* in your study of TM111.

2.4.5 Conclusion

Understanding binary numbers is important for gaining an understanding of how computers compute – that is, using binary switches, but at a tremendous speed. It is also important to distinguish between the units that are used for data storage (bytes) and those used for data transmission (bits). These ideas will occur later in this module and in further studies of computing and IT. In the next section we'll look at how current developments are taking us into the future.

2.5 The future is all around us

Activity 2.23 (exploratory)

Before you start to read this section, think back to yesterday. Spend a few minutes making a brief list of any computers you interacted with over the course of the day, and estimate roughly how many of them there were. I'll return to this shortly.

When asked what a computer looks like, many people today think of a personal computer such as the one illustrated in Figure 2.10 in Section 2.2.2 – a machine equipped with a monitor, a separate keyboard and mouse. They may think of a portable machine such as a laptop computer with a display screen and/or a trackpad, or a tablet or other device with a touch screen (see Figure 2.18). A smaller number of people might first think of a dedicated games console. Again, these are fairly recognisable as computers – at least they still have a type of monitor! Yet although games consoles and personal computers are the most visible and versatile (and most lucrative) kinds of computers, they're not the most common. In fact, you may be surprised to learn that they probably represent less than 2% of the computers in the world!

Figure 2.18 The mobile computers of today

After personal computers and games consoles, you might nominate the apparently omnipresent mobile phone or smartphone as the next candidate in the computer popularity contest. **Mobile computing** allows us to transmit data, voice and video wirelessly across communications networks. Your smartphone includes the facility to browse the web and run games programs, in which case it is certainly a computer – and it has a display screen and a kind of keyboard, too. Many devices have voice recognition software capability to use instead of the keyboard. But even smartphones are not the most common kind of computer. By far the most common computers are those referred to as *embedded*.

An **embedded computer** is one that controls a specific device and that has usually been designed to carry out a highly specialised function. A vast range of devices contain embedded computers, so the chances are that you have interacted with a large number of them today. For example, you may have used a bank card in a chip-and-pin system at the shops, or switched on your microwave oven, or driven your car, or used a swipe card to access a building or service. Did you buy a cup of coffee from an automatic vending machine? All of these examples involve highly dedicated computers – embedded computers that may have neither keyboards nor monitors. However, note that the embedded computers I have described in these examples might not fit everyone's definition of a computer, since user interaction with them is tightly constrained. For this reason, some people prefer to refer to them as *microcontrollers* or *microprocessors,* rather than computers.

As well as the large number of embedded computers that can be found all around you, if you went online to use the web today then you probably interacted with many more computers that you were quite unaware of: computers such as **web servers**, which respond to requests for web pages.

Activity 2.24 (exploratory)

In the light of what you have read since Activity 2.23, return to the list of computers that you interacted with yesterday and update it. Remember that you're only asked for a rough estimate of the number of such computers.

Comment

Here are some of the interactions on my list, but there were lots more.

- I was woken by my digital alarm clock – that's one.
- I drove my car. In doing this, I must have interacted with about 20 computers – they would have been involved in everything from using the accelerator to climate control, changing gear and listening to the radio.
- I bought petrol. There was certainly one computer associated with the pump, not to mention the credit card transaction, which probably triggered several others at my bank.
- I stopped at the ATM for cash – that's at least one – but the ATM must have contacted my bank, so that's two at least, and I'm ignoring all the computers on the network between the two.
- I used my mobile phone – that's probably more than one in itself, but then there are the embedded systems in the phone network to consider and the phone of the person I'm calling… I'm guessing, but maybe 10 or so in total.
- I swiped my ID card to get into the office: a couple there.
- I checked my electronic diary for appointments.
- In my office, I researched material for TM111 on the web, which involved maybe 10 servers for each search and 50 searches – say 500 altogether.

- When I bought lunch, the till was electronic, but I won't count this since I didn't use it directly.
- Oh, then there's my own laptop….

I make that 538 computers in total, but you can see that the numbers are really quite vague. To be honest, I have no idea how many computers I used while carrying out many of the above activities. For example, when I was searching on the web, I could easily have used many times the 500 servers I've quoted.

Activity 2.25 (self-assessment)

Figure 2.19 shows a timeline on which I have identified a few of the more significant developments in computing and related technologies over the last 60 years. Take a little time to add in some of your own, based on your study of TM111 so far.

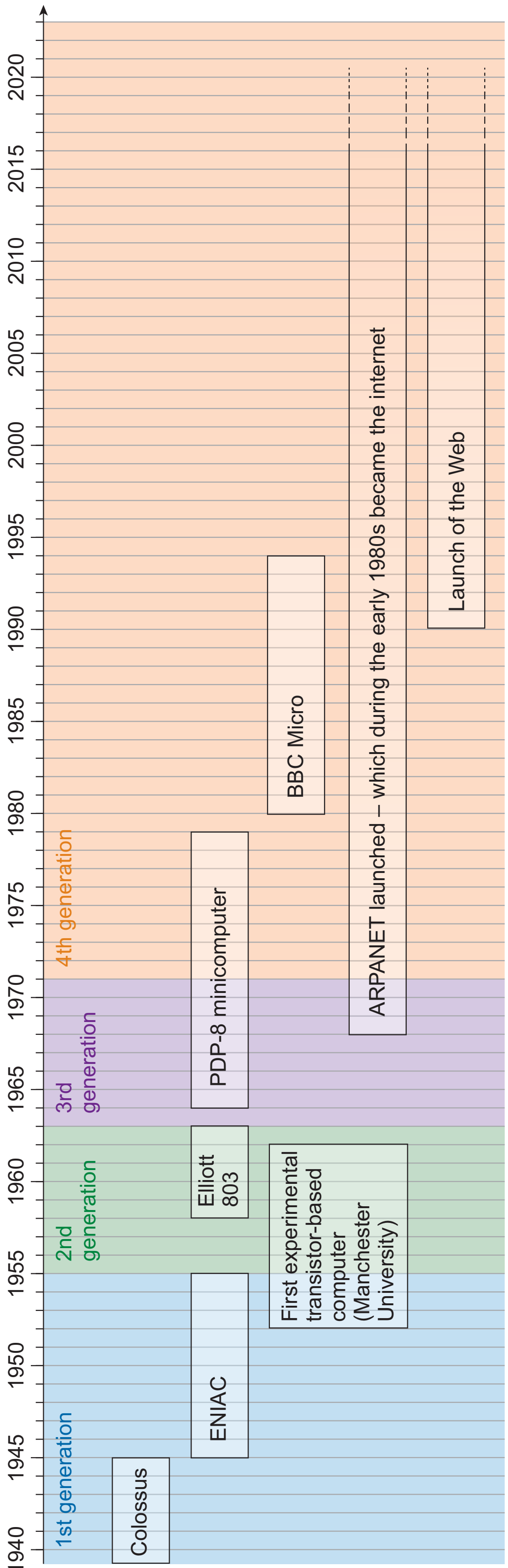


Figure 2.19 Timeline showing some significant developments in computing

2.5.1 The pervasive computer

The examples I've just been considering all add up to one fact: computers are everywhere around us, increasingly influencing all sorts of aspects of our lives. They are also increasingly 'invisible', in that we are often unaware of their presence and indeed of their interactions with each other.

Ubiquitous computing

The term **ubiquitous computing** describes the idea that computers are becoming *pervasive* – that is, they are integrated into the world around us. The term was coined by Mark Weiser at the Xerox Palo Alto Research Center (better known as Xerox PARC). It was there that many of the most significant developments in the field of computing took place, including the mouse and the GUI, laser printing and Smalltalk, the software language behind OUBuild (the programming environment you will use as part of TM111). The following quotation explains what Weiser meant by ubiquitous computing.

> Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives.
>
> *(Weiser, 1996)*

The four generations of computer described in Sections 2.1 and 2.2 were identified using the technologies they were based on, i.e. valves, transistors, integrated circuits and microprocessors. When we talk of ubiquitous computing, we move to a different perspective in which we begin to categorise technologies by our interactions with them. As you have seen already, early computers supported a number of users; subsequently computers became personal; more recently, we each find ourselves interacting with a number of computers, each of which may itself interact with other people and other computers (Figure 2.20). Often we are not even aware of where data is coming from or where it is stored – and more than this, we're often unaware that we are interacting with computers at all. Ubiquitous computing is clearly already with us.

Calm technology is really about technology not being intrusive, thus allowing us to relax (feel calm) in its presence. This doesn't mean the technology has to be invisible, but it shouldn't give (or ask for) more information than is necessary. However, if we don't want our daily lives to be constantly interrupted, then we have to be prepared to allow data exchange to take place – and this may have consequences, not least for data security. Some people would be very happy for technology to be almost invisible, but this does imply a certain loss of control; the technology has to make decisions that would otherwise have been yours.

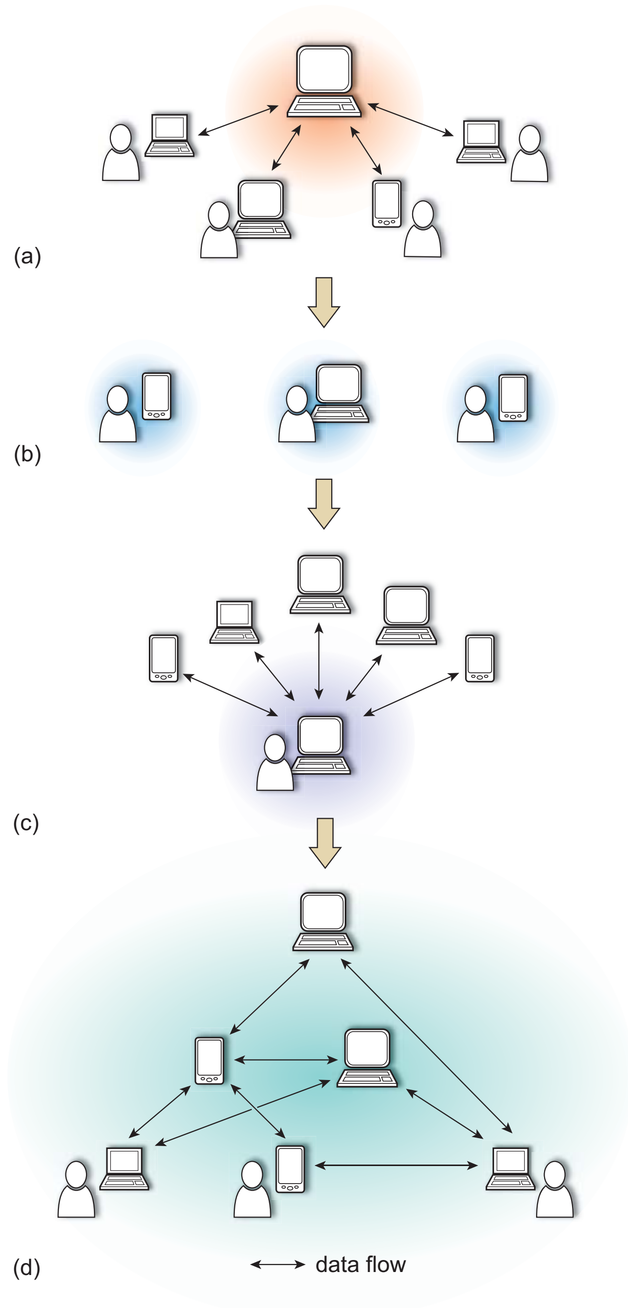


Figure 2.20 The different interactions between people and computers: (a) many people interacting with a single computer; (b) people each interacting with their own computer; (c) individual interacting with many computers; (d) many individuals and many computers all interacting

Cloud computing

You are probably already familiar with 'cloud' computing services such as DropBox and OneDrive for storing and sharing files, such as documents and photographs. As we become increasingly 'mobile' in our use of computers, we find that we need to share information not only with other people, but also between our own devices, both for work and leisure purposes. Individuals and organisations use **cloud computing** services both to share data and for backing up data.

For many businesses and individuals, the financial overheads of computing can be very high; they may need to invest in and upgrade hardware and software that they don't use regularly. For example, take a small publishing company consisting of ten staff, perhaps producing a magazine. The staff may need a range of applications in their work, such as word processing, desktop publishing, databases of their publications as well as accounting packages. They will also need to share documents with each other and work outside of the office, for example when undertaking assignments abroad. At certain times, the company might need to store large amounts of data, while at others it might have much smaller data storage requirements. Finally, some of the software it requires might be used only rarely – spreadsheet applications for a yearly audit, for example.

For these reasons, the cloud seems an excellent option for this company. Storage capacity and processing power can be provided remotely, from the 'cloud' – the internet. Under such a model, the business may pay only for the computing facilities that they use in much the same way as for utilities such as gas or electricity (Figure 2.21). This is known as *utility computing*.

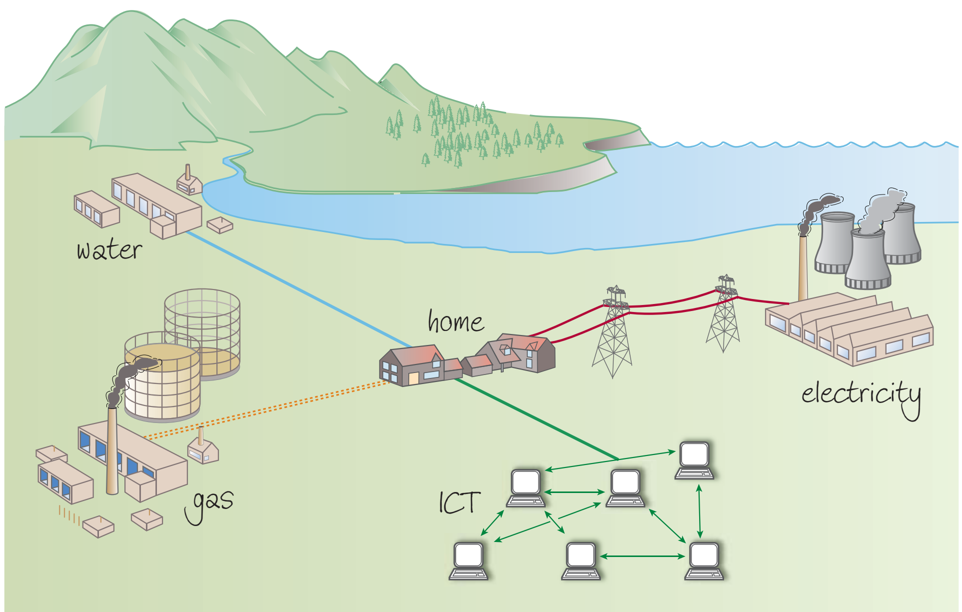


Figure 2.21 Computing as a utility

Certainly, the cloud model offers greater freedom to expand whilst apparently reducing hardware, software and energy costs and increasing reliability. But are there other, hidden issues?

Activity 2.26 (exploratory)

What issues do you think might cause problems or be a cause of concern with regard to cloud computing?

Comment

Here are some issues that I thought of.

- There could be technical problems in the cloud, such as network disruptions, which might make data or services unavailable.
- Data may be insecure. Who would own the data, for example, the company or the cloud provider?

Ethics

I hope you can see from Activity 2.26 that there are some interesting debates emerging about the rights and wrongs of how we use computers and IT. Do you take the view that the state should have the right to look at your information? If so, what about the security forces from another country?

There is an ethical dimension to nearly every technology, whether it is stem-cell research, weapons development or satellite TV. The decisions we make about the application of these technologies depend not only on technical feasibility but also on our values and beliefs. You can probably see already that ethics is a pervasive subject and something that, as a student in this field, you can't ignore.

2.5.2 Conclusion

In this section, I have introduced ideas about more contemporary computers. I discussed ideas about what a computer is and what it looks like. Our 'personal' computers are likely to have some form of display screen and a keyboard or voice recognition software so that we can interact with them. Other computers are 'embedded' in devices that we use on a daily basis. Other forms of computers are 'ubiquitous' in that they are everywhere around us, interacting with each other and increasingly influencing all sorts of aspects of our lives.

Summary

In this part, *The evolving computer*, I started by looking at early computers and what they did. You saw that, over time, computers have become smaller, more numerous, more interconnected and more portable. I considered how, over time, the concept of computing has moved from several people working around one large computer in the early days to that of one person working on their own machine. Now, the concept is of computers being all around us, embedded in all sorts of devices and machines: truly inside anything, anywhere.

I also looked at some aspects of what happens inside a computer. You learned about the fundamental idea of using binary numbers to code digital representations, and how binary numbers can be stored inside a computer.

From your work on this part, you should be able to appreciate, in outline, how various kinds of information can be represented inside a computer. For example, consider a document that contains only text, such as the characters on this page; each one can be represented by a binary number. A document, then, can be represented by a string of binary numbers in a computer.

This section should have helped you to:

- list some milestones in the history of computing, including that of the personal computer
- understand how computers became widely used in business and in the home
- identify potential devices in which computers might be embedded
- explain the concepts of ubiquitous computing, cloud computing and mobile computing
- understand the concept of exponential growth
- work with exponential notation in simple contexts, and with prefixes for large numbers such as kilo, mega and giga
- convert between decimal numbers and binary numbers
- recognise the difference between units used for storage – bytes – and those for transmission – bits.

In terms of your study skills, you have made notes relating to the printed texts and on an audio recording. You have also practised digital and information literacy (DIL) skills, such as sharing an image of your computer with your fellow students.

Answers to self-assessment activities

Activity 2.3

(a) The power consumption of the electric iron is 1 kW = 1 × 1000 W = 1000 W.

The power consumption of the electric kettle is 2 kW = 2 × 1000 W = 2000 W.

(b) The electric kettle, fire and iron use the most power and are all designed to produce heat.

(c) The power consumption of ENIAC was 170 kW. The power consumption of an electric fire is 2 kW, so we simply divide the former by the latter: 170 kW ÷ 2 kW = 85.

So 85 electric fires could be run on the power required for ENIAC.

Activity 2.5

(a) Generation 1: valves.

Generation 2: transistors.

Generation 3: pre-microprocessor integrated circuits.

Generation 4: microprocessors.

(b) Each successive generation of computers required less energy for a given task, and became faster, smaller and cheaper to use and to manufacture in large numbers.

Activity 2.8

(a) 1965 was during the third generation (which was founded on the invention of the integrated circuit).

(b) The move from the third to the fourth generation was led by development of the microprocessor – an integrated circuit containing the CPU.

Activity 2.10

Consider the number of grains on each square.

- The first square contains 1 grain of rice.
- The second square contains 2 grains of rice: 2 multiplied by 1, or 2.
- The third square contains 4 grains of rice: 2 multiplied by itself twice, or 2 × 2.

- The fourth square contains 8 grains of rice: 2 multiplied by itself three times, or $2 \times 2 \times 2$.
- The fifth square contains 16 grains of rice: 2 multiplied by itself four times, or $2 \times 2 \times 2 \times 2$.

By now, you can probably spot the pattern. If we consider the nth square, where n is any number from 2 to 64, the number of grains of rice on that square is 2 multiplied by itself $n - 1$ times.

So the number of grains of rice on the 21st square is 2 multiplied by itself 20 times.

Activity 2.11

$16 = 8 \times 2$, so 16 years is 8 lots of 2 years.

After 8 lots of 2 years, the number of components in an area that originally contained one component would be 2^8.

$2^8 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 256$.

So the answer is 256 components.

Activity 2.12

$2^6 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 64$.

$2^9 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 512$.

$10^3 = 10 \times 10 \times 10 = 1000$.

$10^4 = 10 \times 10 \times 10 \times 10 = 10\,000$.

$8^3 = 8 \times 8 \times 8 = 512$.

You may have noticed that 2^9 and 8^3 have the same value. This is because $8 = 2 \times 2 \times 2$, so $8 \times 8 \times 8$ is the same as $(2 \times 2 \times 2) \times (2 \times 2 \times 2) \times (2 \times 2 \times 2)$.

Activity 2.13

(a) $10^5 = 10 \times 10 \times 10 \times 10 \times 10 = 100\,000$.

(b) The pattern is that 10^n is 1 followed by n zeros.

(c) 1000 000 is 1 followed by 6 zeros. So $1000\,000 = 10^6$.

Activity 2.14

There are 8 three-bit binary numbers in total: 000, 001, 010, 011, 100, 101, 110 and 111.

Activity 2.15

I'd be able to form $2^8 = 256$ binary numbers.

Activity 2.16

The next column weighting is two times the value of the last column weighting, i.e. 2×8 or 2^4 or 16.

The column weighting after that is two times this last value, i.e. 2×16 or 2^5 or 32.

The column weighting after that is two times this last value, i.e. 2×32 or 2^6 or 64.

Activity 2.17

(a) For 111_2, I have the following from the table:

- $1 \times 4 = 4$
- $1 \times 2 = 2$
- $1 \times 1 = 1$

So $111_2 = (4 + 2 + 1)_{10} = 7_{10}$.

(b) The decimal equivalents are:

(i) $10_2 = 2_{10}$

(ii) $10001_2 = 17_{10}$

(iii) $11000_2 = 24_{10}$

(iv) $11111_2 = 31_{10}$

Activity 2.18

The next number in the series is 256, which would exceed 157, so I stop at 128.

Activity 2.19

128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
$178 - 128 = 50$		$50 - 32 = 18$	$18 - 16 = 2$			$2 - 2 = 0$	
1	0	1	1	0	0	1	0

The table shows that the decimal number 178 is equivalent to the binary number 1011 0010.

Note that it is important to write down the final 0. The binary numbers 1011 0010 and 101 1001 are not the same. (You can check this by working out the decimal equivalent of 101 1001 – it is not 178.) This is just as with decimal numbers where, for example, the numbers 23 and 230 are not the same.

Activity 2.20

(a) 1100 0111

(b) 100 0000

(c) 1 0100 1000

(d) 11 1110 1000

Activity 2.22

(a) $512 \times 1048\,576 = 536\,870\,912$ bytes.

(b) $536\,870\,912 \times 8 = 4\,294\,967\,296$ bits.

Activity 2.25

Figure 2.22 shows the detailed timeline I drew up.

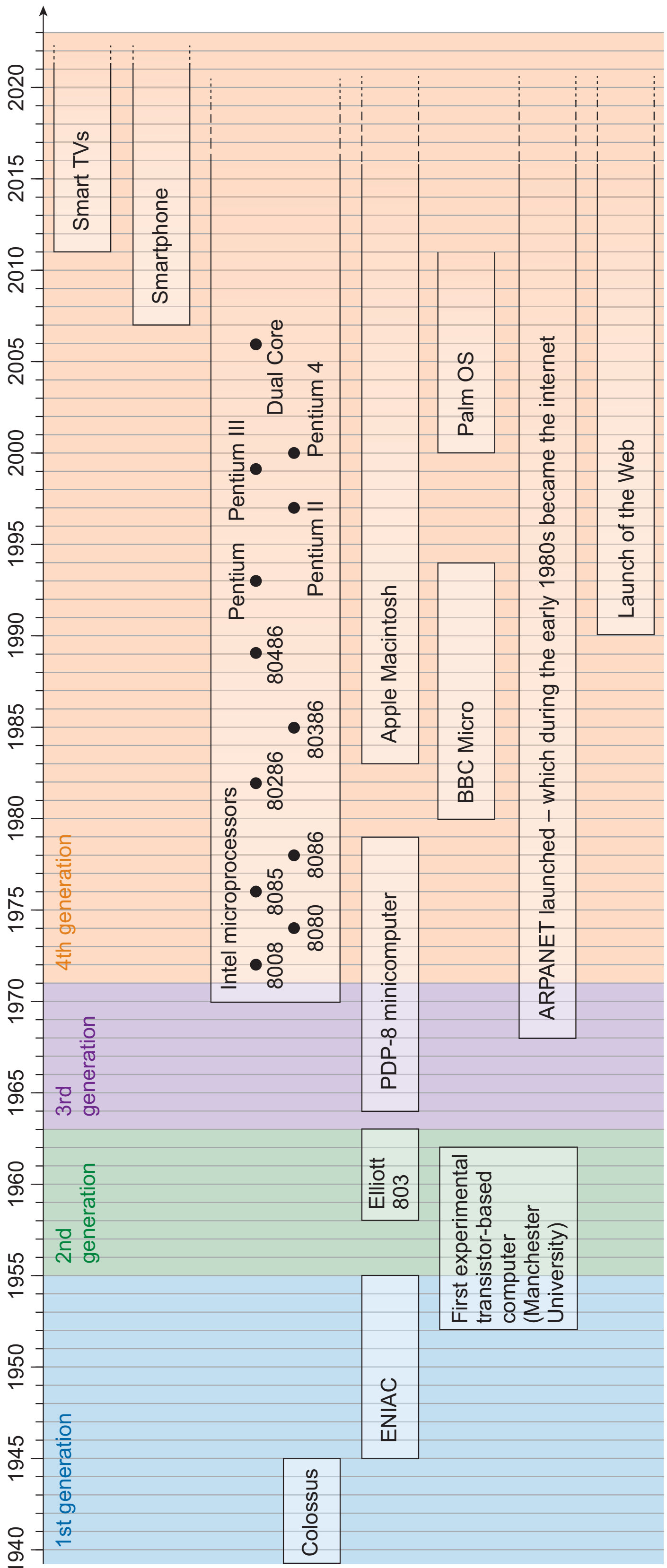


Figure 2.22 A more detailed version of the timeline from Figure 2.19

References

Moore, G. E. (1965) ‘Cramming more components onto integrated circuits’, *Electronics*, vol. 38, no. 8, 19 April.

Weiser, M. (1996) *Ubiquitous Computing* [Online]. Available at www.ubiq.com/hypertext/weiser/UbiHome.html (Accessed 10 March 2017).

Acknowledgements

The TM111 module team acknowledges the contribution of Tony Nixon to the production of these materials.

Grateful acknowledgement is made to the following sources.

Figure 2.1. © https://commons.wikimedia.org/wiki/File:NAMA_Machine_d%27Anticyth%C3%A8re_1.jpg This file is licensed under the Creative Commons Attribution-Noncommercial-ShareAlike Licence http://creativecommons.org/licenses/by-sa/3.0

Figure 2.5a. Taken from: http://www.wylie.org.uk/technology/semics/Ediswan/Ediswan.htm

Figure 2.5b. Taken from: http://www.columbia.edu/cu/computinghistory/608.html

Figure 2.7a. http://www.antiquetech.com/wp-content/uploads/2013/09/4004chip.jpg

Figure 2.7b. Taken from: http://www.antiquetech.com/wp-content/uploads/2013/09/C4004Gray.jpg

Figure 2.7c. Taken from: http://www.techiwarehouse.com/cat/71/Technological-Advances

Figure 2.9. Taken from: http://img5.wtoutiao.com/?url=http%3A//mmbiz.qpic.cn/mmbiz/HxZ9OsVG03bzmh3WFsZfvVIXfPfS6icb2gfxcZBDiafntvOc6LtrQzMsiaIx4x-C58eIwA94fAXOtpicdv5a74n3kvg/0?wx_fmt=jpeg

Figure 2.18. Taken from: https://www.lg.com/tr/images/mobil-iletisim/features/lg-google-nexus-4-android-beam.jpg

Every effort has been made to contact copyright holders. If any have been inadvertently overlooked, the publishers will be pleased to make the necessary arrangements at the first opportunity.

Part 3 Digital media

by Soraya Koudari Mostéfaoui and Elaine Thomas

Contents

Introduction

What do we mean by 'digital media', as used in the title of this part? The term 'digital media' includes both the tools we use to create media such as text, images, audio and video, and also the idea of communicating and sharing through the internet and the web.

Technologies have revolutionised the production and distribution of digital media content as well as its consumption. Similarly, education has undergone tremendous changes, with advanced technologies now commonplace in supporting learners from early years through to higher education. Whether in an educational or training context or in a work-based setting, the use of audio and video offers a richer and more interactive environment for educators or other presenters to express messages.

Moreover, as Marshall McLuhan famously said, 'the medium is the message' (McLuhan, 1964). This phrase has been quoted in many different contexts, but essentially it means that the way in which we are entertained or receive explanations can have as profound an impact as the actual content which is being communicated. We are now so used to audio and video content being readily available – whether online, via mobile phones or via more traditional TV and radio – that most of us now have an expectation that our learning and entertainment will include such technologies, and this has changed the way in which we learn and are entertained.

This part takes quite a different approach from the previous two parts. You will be learning practical skills that will enable you to produce a short audio recording and, although there is some theoretical background material, most of your study time will actually be spent working on practical tasks and activities. You will learn how to use an audio editing tool to manipulate sound.

The software tool

Audacity is freely and openly available, but you'll find that a particular version of it is provided on the module website and it is recommended that you use this. The techniques used in this part are essential, and thus should be available in future versions of the software. Indeed, these techniques are so essential that they are features provided by pretty much any sound-editing tool, openly available or not.

A note on privacy and data protection

As a rule, in this part, you will be sharing your digital media with other TM111 students and tutors in OpenStudio, which is only accessible within this module. However, digital media are commonly shared on the web, so it is important to bear in mind issues of privacy and its 'legalese' counterpart, data protection.

You may wish to upload materials – such as family snapshots – to a website that you want to share only with family and friends, so you need to make sure that you can restrict access exclusively to the groups of people you want. The concern here is not only privacy but also copyright and intellectual property rights (IPR), which are (normally) covered under a site's terms and conditions.

In a nutshell: if you upload, for example, a photograph to a website, who 'owns' the image, who has the 'right' to do things with the image and, specifically, which things would be allowed? If you upload a photo of your child to a site on an open-access basis, you may be surprised later on to find that photo on someone else's website.

Of course, most material on the web can be grabbed, copied and modified (though there are some ways to make this harder, it isn't possible to prevent it entirely), but this is often illegal. Privacy and copyright/IPR are, indeed, areas of great ethical and legal concern, partially because of the amazingly quick pace at which new technologies appear.

A media database of images and sounds has been provided on the module website which you can use for the TMA question associated with this part. Other files are made available under a Creative Commons license which allows people to use copyrighted work under specified conditions. As you'll see, in the supplied audio files on the module website, I've used two classes of sounds – samples obtained from the web and voice recordings made for this module – and I've restricted my work to manipulating and mixing these samples.

Activity 3.1 (exploratory)

To begin your study of this part, take a few moments to consider the digital media you already use as a 'consumer'. Jot down your thoughts on, and perhaps list, a couple of tools that you use to create media yourself.

Comment

I use a lot of audiovisual media in my daily life. As a 'consumer' of media, I watch television, listen to the radio, listen to music and look at pictures online. I also look at photographs and videos shared by my friends and colleagues on social networking sites. As a media 'creator', I like to take photographs on my camera and share them electronically later. On short visits to places and events, I frequently use my smartphone to record some of the things I see and hear and post directly to social networking sites.

Representing the real world in digital form 3.1

In using and creating digital media, we are often representing part of the real world in a digital form that can be stored, transmitted or manipulated using information and communication technology.

There are two major ways of representing information from the real world. The first has come to be known as **analogue** representation. You could say that we live in an analogue world with an infinite range of sounds, colours, touches, tastes and smells that we can perceive through our senses. In analogue representations, the representation is a direct counterpart (analogue) of the real-world aspect that we are trying to capture. For example, an analogue clock displays the time using a pair of hands that sweep around the face pointing at a series of numbers while a digital clock uses only numbers in a digital representation. Let me try to make this concept clearer using the example of temperature.

Temperature is analogue and, by its nature, can take on any value within a given range. So when temperature is measured by a traditional mercury- or alcohol-in-glass thermometer, we have a **continuous**, analogue representation. (The level of the liquid column in the glass tube is analogous to the temperature.) Just as the temperature itself can take on any value in a given range, so can the level of the liquid in the thermometer.

(a) (b)

Figure 3.1 (a) A traditional analogue thermometer and (b) a digital thermometer

There is, however, another way of measuring temperature. We don't have to use liquid in a thermometer; a common alternative method is to use a digital thermometer, as illustrated in Figure 3.1(b). In this instance, a sensor is used to convert the temperature into an electrical **voltage**. At this stage, the representation is an analogue voltage that varies – but to display the temperature, it is normal to use an electronic circuit to convert the analogue voltage into a digital representation, as a numerical value of degrees Celsius or Fahrenheit. This means that the values are converted into **discrete** values.

For a room thermometer, such a display would probably be to the nearest degree. The display cannot show just any value, and a temperature of, say, 25.8 °C would appear as 26 °C. Even a more accurate digital thermometer designed for scientific or medical purposes, for example, can only use a finite number of digits in a display. Figure 3.2 shows two temperature plots on the same axes: one a plot of a continuously varying temperature, and the other the corresponding series of discrete values that might come from a digital thermometer. You can see from this figure that the two plots are quite different.

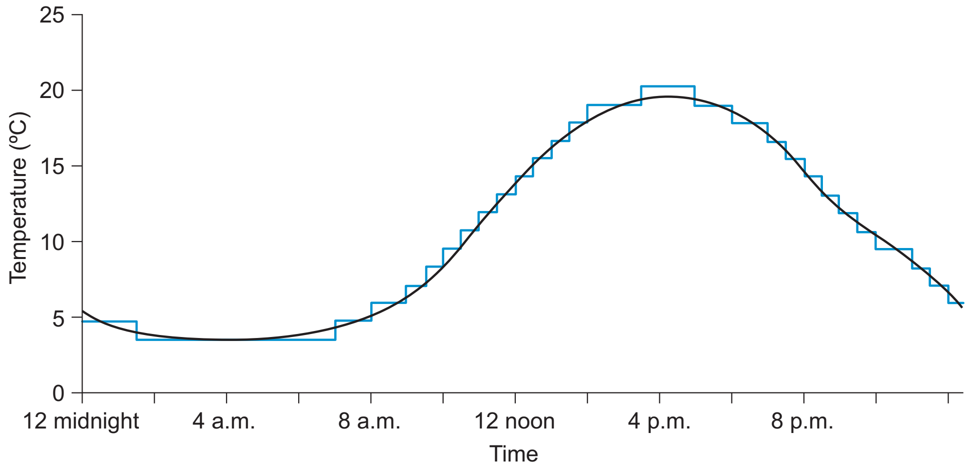


Figure 3.2 A graph showing a continuously varying temperature along with the discrete values plotted against time

It's important to note that many quantities are essentially discrete, in that there is only a fixed range of values they can take. One example of a discrete quantity is the number of people in a room; people are counted in whole numbers: one person, two people, etc. As a rough guide, a discrete value is something that can be counted, whereas continuous data is measured.

Activity 3.2 (self-assessment)

Which items in the following list are fundamentally analogue and which fundamentally discrete?

(a) The amount of heat from a fire

(b) The energy of a star

(c) The pressure of the atmosphere

(d) The price of petrol

(e) The size of the audience at a play

(f) The speed of a car

One key factor in the rise of digital communication and data storage is that almost any analogue quantity (such as a sound pressure wave or the colour of a light) can, in principle, be represented digitally. This, in turn, means that the representation can be stored and manipulated by a computer, with the values of the digital representation being 'coded' as binary numbers that can be stored.

Activity 3.3 (self-assessment)

What are the essential differences between 'analogue' and 'digital' representations?

3.1.1 Representing images

In principle, digital codes can be used to represent any type of information. Let's now investigate images on a computer (or mobile phone) screen, an increasingly important aspect of our 'digital world'.

Digital images

On a computer screen, images are generated by dividing the display into a large number of tiny units called **pixels** (from 'picture elements'). Each pixel can, in general, be displayed in a number of different colours and levels of light intensities or brightness. The greater the number of pixels used for a given size of display, the more detail can be displayed and the higher the quality of the image. To begin with, we'll look at the representation of black and white images, and come back to colours later.

Let's consider an artificially small-scale example: a display using eight rows of sixteen pixels. In Figure 3.3 I have shown how this system could display a 'triangle'.

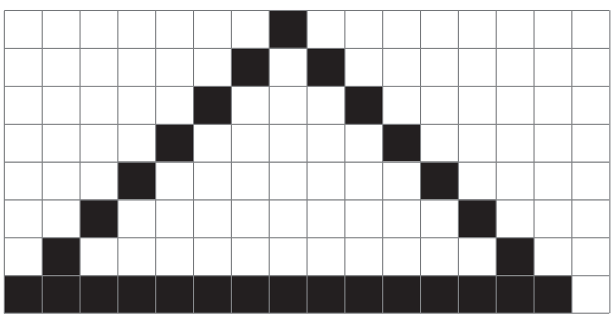


Figure 3.3 Black-on-white 'triangle' on a 16 × 8 pixel display

Activity 3.4 (exploratory)

What do you notice about the representation of the three sides of the 'triangle'?

Comment

The horizontal side is a perfect straight line, since all the pixels are lined up along the row. The other sides have a 'staircase' appearance, and only approximate a straight line. You may well have noticed this effect on old mobile phone displays, which often used only a small number of pixels compared with, say, a computer display or the screen on a contemporary smartphone.

Next time you use your computer, look at a photograph in an application such as Paintbrush on a Mac or Microsoft Paint or Paint 3D, and 'zoom in' as far as you can (an option under the View menu) to examine it in detail. Figure 3.4 shows a photograph and a section that has been 'pixelated' by zooming in. The number of pixels in the 'zoomed in' section hasn't changed, but each pixel has expanded in the display so that it is possible to see each one as a discrete square.

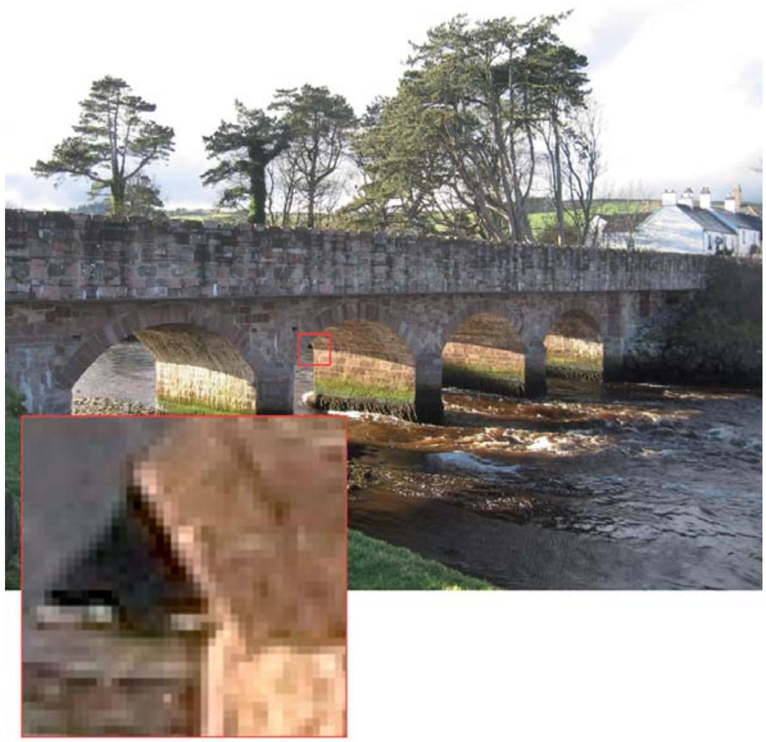


Figure 3.4 A photograph of the Antrim Coast – a section of the photograph has been enlarged using the zoom facility to illustrate the pixel effect

I noted above that a defining feature of digital techniques is the representation of the information by codes. Let's look now at how that can be done for the simple triangular image of Figure 3.3.

Information is stored, processed and transmitted digitally in a great variety of ways. One of the most common is binary representation. You learned in Part 2 that just two symbols or 'states' are used in binary representation. For example, storing data in solid-state storage media, such as a hard disk or a memory stick, involves tiny electronic switches (transistors) that are either off or on. For transmission over optical fibres, the two states are the presence or absence of a short pulse of laser light. The two states of binary representation are written 0 and 1. So, for example, the 0 could be represented by an absence of a pulse of laser light, and the 1 by a pulse of such light. For most of this module, we'll not be concerned with precisely what physical quantities are used for the binary 1s and 0s, but we shall be concerned with how to use binary coding for the transmission, processing and storage of information.

Let's get back to our image of a triangle. If we represent a white pixel by a binary 0, and a black pixel by a 1, we can code the image as follows:

```
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
```

We've now converted the image into a set of binary 1s and 0s that we could store or transmit using any digital technique at our disposal. Provided we always use the same conventions (for example: 0 = white; 1 = black; display of 16 × 8 pixels read from top left to bottom right), we can reconstruct the image pixel by pixel.

Activity 3.5 (self-assessment)

Figure 3.3 is a black triangle on a white background. How would you code a similar white triangle on a black background using this simple scheme?

This is a very simple illustration of the enormous potential of digital processing. Once we have coded the information in digital form, all sorts of possibilities for manipulating the information arise. For example, by means of a simple instruction, we can make overall changes to an image, such as changing black to white – or one colour into another, as you will see later. This technique lies at the heart of many more complex processing functions – not only for images, but also for text and sound.

The display system described so far is simple, but quite restricted. By increasing the number of pixels, we can improve the amount of detail the image contains – that is, its resolution – so as to get high-quality, black-and-white images.

The screen of the laptop I am using allows a range of options from 2560 × 1600 pixels down to 1024 × 640 pixels. Screens that use 3840 × 2160 are available now, but screens using even larger numbers of pixels are being developed. Increasing the resolution means that more pixels must be coded and stored. To get an idea of the numbers of pixels involved in a practical situation, try the following activity.

Activity 3.6 (self-assessment)

(a) What is the total number of pixels in the display for the two options for the laptop just mentioned?

(b) What is the ratio of the two totals (the higher divided by the lower)? You could use a calculator to work this out.

But what about improving the representation by including shades of grey, or different shades of colour? We can extend our binary representation of each pixel so that, instead of using just one bit, we use two bits for each colour. We now have four possible codes. for example:

00 white

01 red

10 blue

11 black

Binary coding can be extended to represent an increasing range of colours. For eight colours, for example, we need to use three bits for each. The actual allocation of colours to codes is arbitrary, but there has to be an agreed standard to ensure interoperability – in other words, so that software produced by different companies will result in the same colours being displayed on computers from different manufacturers. (I'll have a lot more to say about standards in later parts.)

The following scheme is used in practice for situations where each colour is represented by three bits:

000	black
001	red
010	green
011	yellow
100	blue
101	magenta
110	cyan
111	white

(In colour optics terminology, magenta is a mixture of red and blue light; cyan is a mixture of blue and green.)

Perhaps you can see the general pattern in the codes used for the colours. To get the eight codes, we take the four colour code system mentioned earlier:

00

01

10

11

Then we put a binary 0 in front of each, to generate four new, 3-bit, codes:

000

001

010

011

Next we put a binary 1 in front of the four original codes to generate another four new codes:

100

101

110

111

A set of a given number of bits used for data representation (in general, not just for colours) is called a **binary word**: in this example, we have eight, 3-bit, binary words to represent the eight different colours.

Activity 3.7 (self-assessment)

How long must a binary word be to represent 16 different colours? Write out the 16 code words.

Coding images

Any coding scheme that allocates a binary code to each pixel in order to define its colour and brightness is called a **bitmap**. The term is usually restricted to coding schemes that code these values directly – normally to computer files with the extension .bmp that can be displayed by a wide range of applications.

File formats are agreed-upon, standardised ways of organising and storing the binary data that represents an image in your computer. Of course, there are many ways of organising this data, so there are quite a number of formats available, some of which are specific to the operating system. However, a major issue relating to file formats is the resulting size of the file. The problem is two-fold: large files require a lot of storage space, and they are also more laborious to transfer across storage media for the purposes of backup and (as far as the web is concerned) transmission.

When a bitmap displays a colour image, such as the scene in Figure 3.4, there are several shades of gradation in colours and brightness. In this case, each pixel in the bitmap might have 16, 24, or 48 bits of information associated with it. The more bits, the greater the resolution of the bitmap and, therefore, the larger the file. Because bitmap files are relatively large, they're not the most appropriate choice of format for images to be viewed on the web. For example, the size of the bitmap file for the image in Figure 3.4 is 1.8 MB.

A very different approach to coding images is known as **vector graphics**. In a vector graphic representation, instead of storing information about each pixel, information is stored about the different shapes in the image. Vector graphics are most appropriate where the image is made up of regular shapes, as is the case in many web or computer game graphics.

For example, a straight line can be defined simply by its starting and end points, or a circle by its centre and radius. In this way, file sizes can be kept much smaller than a bitmap, provided the complete graphic consists only of the shapes allowed within the vector graphic system. It is also easy to produce simple animations of such shapes. For images such as photographs or motion pictures, however, the vector graphics approach is not effective.

3.1.2 Real-world implications

Now that you've seen some of the basics of binary representations of images, let's have a look at some real-world implications. Consider, for example, the following general questions. I don't want you to try to answer them now; I'll ask you to do some calculations of actual values in a moment.

1 How large is the data file for an image?
2 How long would it take a user to download such an image from the internet?

Note: I'm using a calculator for the next piece of work!

Earlier I mentioned that some bitmap formats use 16 bits for each pixel. Currently, a medium-priced smartphone has a screen resolution of 640 × 960 pixels. How much data is required to display an image?

$640 \times 960 = 614\ 400$ pixels in total

$16 \times 614\ 400 = 9\ 830\ 400$ bits

As you saw in Part 2, a byte is 8 bits or 2^3; a kilobyte (KB) is 1024 bytes or 2^{10}; and a megabyte (MB) is 1024×1024 or $2^{10} \times 2^{10}$, which is 2^{20}.

Therefore, to convert bits to megabytes, the number of bits is divided by 2^3 to convert it to bytes, and the resulting answer is divided by 2^{20}, as follows:

$9\,830\,400 \div 2^3 = 1\,228\,800$

$1\,228\,800 \div 2^{20} = 1.171875$ megabytes (MB)

Hint: On a scientific calculator, the 'power' key might have the symbol x^Y.

Alternatively, you can calculate this without the scientific notation:

$9\,830\,400 \div 8 \div 1024 = 1.171875$ megabytes

If another file in bitmap format is 3 MB in size, how long will it take to download the file from the internet if my internet connection is running at 14.1 Mbs? Notice the difference in the units: the file is 3 mega*bytes*, while the internet speed is 14.1 mega*bits* per second. In order to calculate the time taken to download the file, we must first convert the values to bits.

Therefore, the file size can be expressed as

$3 \times 2^{10} \times 2^{10} \times 8$ (or $3 \times 1024 \times 1024 \times 8$) $= 25\ 165\ 824$ bits.

The internet speed is $14.1 \times 1000 \times 1000 = 14\ 100\ 000$. Using the formula

time = size ÷ speed

gives

time = 25 165 824 ÷ 14 100 000 = 1.78481021276596 seconds = 1.78 (to two decimal places)

Activity 3.8 (self-assessment)

A common bitmap format for displaying reasonably high-quality digital images is to use 24 bits for each pixel.

(a) How many pixels are required for 1024×768 screen resolution? How many bits of data does this require for each image?

(b) My internet connection provides an upload speed of 2.3 Mbs. How long will it take to upload such an image?

3.1.3 Conclusion

In this section, I've discussed how digital codes are used to represent images. You have carried out some numeracy activities to explore the implications of file formats in the storage of data and for transmitting data across the internet. In the next section, I'll be talking about how sound is represented digitally, and you will be carrying out some practical exercises on the processing of sound.

3.2 Sound

In this section, we will be considering sound – what it is, how it can be converted to and from a digital representation, and how it can be manipulated and stored in this electrical form. You will use some of the skills you have gained through studying this part of Block 1 in the production of your short audio file.

I will start this section with a simple practical activity to help you familiarise yourself with Audacity. Then I will explain some key concepts about sound which will help you to understand the next section on the recording and editing of sound. In order to help you understand the principles I will be introducing, you will be asked to carry out a number of practical activities using the module's sound editor, Audacity. It is highly recommended that you carry out all these activities at the time you meet them in the text, which means that you should try to study this part of Block 1 when you have your computer to hand.

3.2.1 Using Audacity

Audacity is a sound recording and editing program that is available for computers using Windows, Macintosh operating system (macOS) or Linux operating systems. The program allows you to load, record, save and play sound using a variety of different file formats, to look at the waveform of sounds, to analyse sounds and generally to manipulate (i.e. to edit) sounds. Before we start our investigation into sound, I would like you to get familiar with the basics of this program so you can start to use it straight away.

Activity 3.9 (practical)

A link to the Audacity software is provided on the module website. You will find it easier to open this in a new tab or window. Download the Audacity installer to your machine, then run it and follow the instructions on screen. Once you've finished, close the other tab/window and return to this page.

As mentioned earlier, this won't be the latest available version of Audacity that you can get from the developers (which you can get by going to www.audacityteam.org/download), but it is the version which the study material has been based on and hence is the version used to create and test this part's practical activities.

You may find it helpful at this point to view the screencast 'Introducing Audacity' available on the module website.

To start getting you familiarised with the basic functionality of Audacity, let's take a look at the tool and load up and play a sound file.

Activity 3.10 (practical)

In this activity you will load and play a sound file in Audacity and have a brief look at the program's operation. You will need loudspeakers or headphones to complete this activity. All operating system versions of the program operate very similarly, but where there are differences, the versions for a Windows computer and a Mac computer will be given. It is likely that the Linux version will operate very similarly to the macOS version, but this version has not been tested with the activities in this part of the block.

1 First make sure your headphones or loudspeakers are connected to your computer, then run the Audacity program, and the Audacity window should appear. There are a number of menu items along the top, and there is a large grey blank window where visual representations of sound files will be displayed when they are loaded. In between, there are various toolbars containing a familiar set of transport controls (play, stop, record, rewind, etc.) and some editing and level controls/indicators.

2 Use `File > Open` to navigate to and open the sound file associated with this activity, which is `Activity_3-10.wav`.

   Note: The first time you open an Audacity file a warning box will appear saying '`When importing uncompressed audio files you can either copy them into the project, or read to them directly from direct current location (without copying).`' Select the '`Make a copy of the files before editing (safer)`' option and also check the box saying '`Don't warn again and always use my choice above`' and click `OK`.

   When `Activity_3-10.wav` opens, you should see a 'picture' of the sound it contains appear in the grey window. This is the waveform of the sound track. Figure 3.5 shows a labelled version of the Audacity window with the waveform for Activity 3.10.

   Click on the play button and you should hear the sound. If you do not hear any sound, then first check that your loudspeakers/headphones are connected and your computer's various volume controls are turned up. If there is still no sound, then you may need to change the default settings: select `Edit > Preferences` (Windows) or `Audacity > Preferences` (macOS), make sure that the appropriate playback device is selected, click on `OK` and see if the sound does now play.

3 When the sound is playing, notice that a vertical cursor moves along the soundtrack, indicating the point in the sound that is being played. Notice also the sound level bars at the top show the volume of the sound being played. The file is a mono (single-channel) sound, so the left and right sound level bars will both indicate the same level. Notice also that the level bars show the maximum sound level as a separate line that remains for about 3 seconds.

4 On the left of the soundtrack, you will find controls to mute the track, to solo out the track (i.e. to play just this track when there is more than one track open), to adjust the balance of the sound from the track between the left and right loudspeakers, and to adjust the sound level of the track. Play the sound again and experiment with these controls. You can shift-click on

the play button (or press shift-space) to play the sound continuously; just click on the stop button or press space to stop playing.

5 A useful shortcut is to remember that pressing the spacebar will start/stop playing of the sound. There are a number of other keyboard shortcuts which you can find detailed in the `Keyboard` tab of Audacity's `Preferences` section (see Step 2). You can also change or add your own shortcuts there.

6 Finally, you can zoom in horizontally to the sound by using the magnifying glass at the left of the toolbar section and clicking within the waveform 'picture'. To zoom out again, right-click or shift-click. There are also menu items (in the View menu), keyboard shortcuts and separate zoom buttons to do this as well. Hover over a button to reveal its function. To enlarge the vertical scale of the waveform, you can drag the bottom of the track window downwards, or you can also zoom in vertically by using the magnifying glass and clicking in the amplitude scale just to the left of the 'picture' section (the lighter grey section containing the numbers 1.0, 0.5, 0.0, –0.5 and –1.0).

   Try making some of these adjustments now to start getting familiar with zooming in and out, as you will need to do this a lot in later activities. You might also want to adjust the size of the complete Audacity window either by maximising it (if it is not already maximised) or by dragging the bottom right-hand corner.

7 When you have finished experimenting, exit from Audacity; there is no need to save any changes.

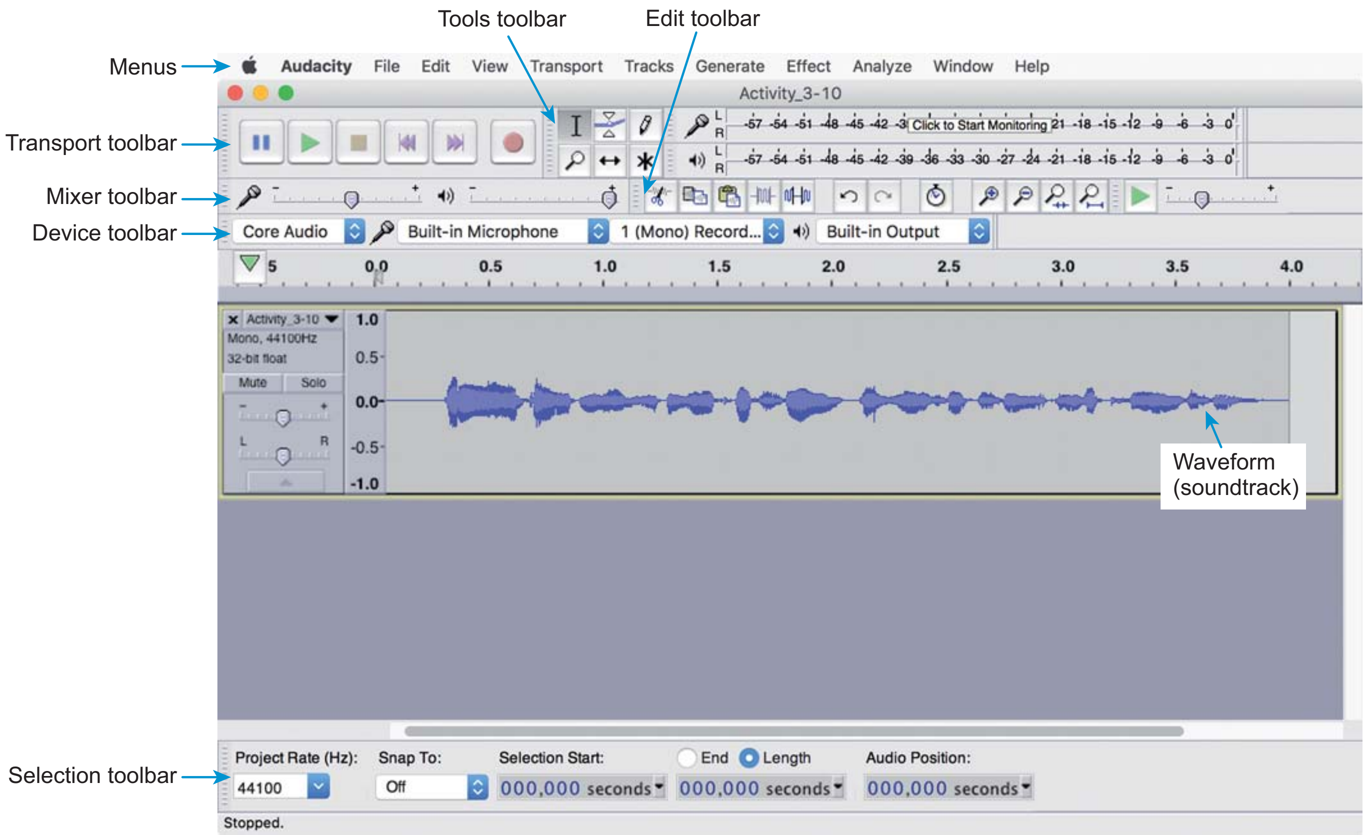


Figure 3.5 Audacity with a sound file opened

We can now start our exploration of sound. Along the way you will be using Audacity at various points – sometimes just to play sound files, and at other times to get practical experience of manipulating sound.

3.2.2 What is sound?

The title of this section poses the question: What is sound? This may seem a straightforward question, but in fact sound is a rather more complicated thing to pin down than you might think on a first analysis. So let's start our exploration of sound by looking at what it is in physical terms.

Physics of sound

Sound is another analogue feature of the world. If you shout, hit a piano key or drop a plate, then you set particles of air vibrating – and any ears in the vicinity will interpret this tremor as sound. The sounds we hear generally consist of small rapid movements (changes, fluctuations) of the atmospheric air pressure that surrounds us. Sound can also be transmitted through other media, such as water, so not all sound consists of fluctuations in air pressure. However, for the purposes of this discussion I shall confine myself to sound in air.

Sound is also reflected when it meets a boundary in the medium it is travelling. The amount of sound that is reflected and the amount that travels into the new medium depends on the medium in question and the composition of the sound.

The fluctuations in air pressure are caused by the air molecules bunching together or becoming more widely separated. They spread outwards from the source through the surrounding air, becoming gradually weaker and eventually dying away completely.

For a listener in the vicinity of the sound source, the pressure variations act on the listener's hearing system, causing the eardrum to move in sympathy with the source of the pressure variations. The movements of the eardrum are detected by the hearing system (shown in Figure 3.6) and are interpreted by the brain as sound. In people with hearing impairment, this system and/or the nerves have been damaged in some way.

An important point to remember here is that sound needs a medium through which to travel – air molecules in this case. If there are no molecules (i.e. there is a vacuum), pressure waves cannot be set up and so sound cannot travel. As the catchphrase for the film Alien warns, 'In space, no one can hear you scream' (Alien, 1979).

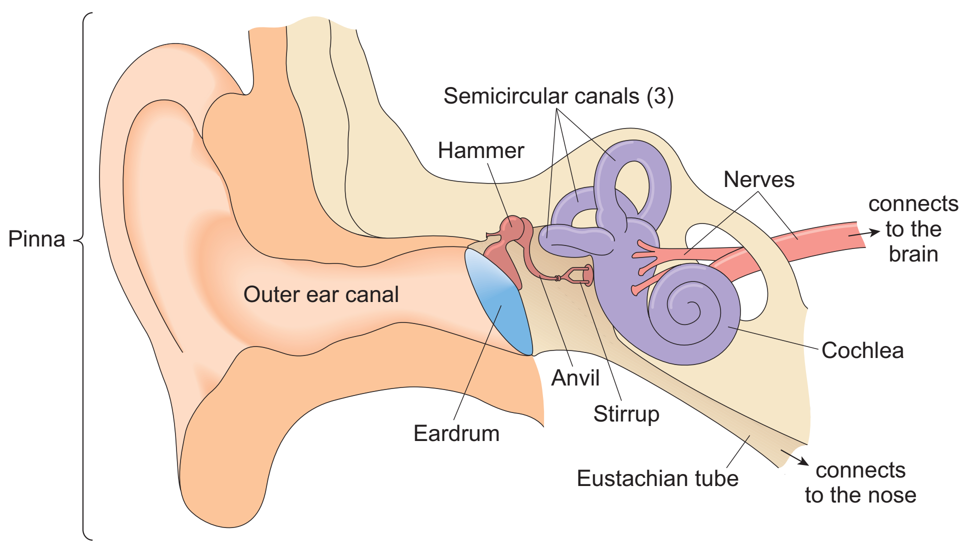


Figure 3.6 Human hearing system

Sound waves

Probably the purest sound you can make is by vibrating a tuning fork. As the prongs of the fork vibrate backwards and forwards, particles of air move in sympathy with them. One way to visualise this movement is to draw a graph of how far an air particle moves backwards and forwards (we call this its displacement) as time passes. The graph (showing a typical **waveform**) will look like Figure 3.7.

Our particle of air moves backwards and forwards in the direction the sound is travelling. As shown in Figure 3.7, a cycle represents the time between adjacent peaks (or troughs), and the number of cycles completed in a fixed time (usually a second) is known as the **frequency**. The **amplitude** of the wave (i.e. maximum displacement – see Figure 3.7) determines how loud the sound is; the frequency decides how low- or high-pitched the note sounds to us. Note, though, that Figure 3.7 is theoretical; in reality, the amplitude will decrease as the sound fades away.

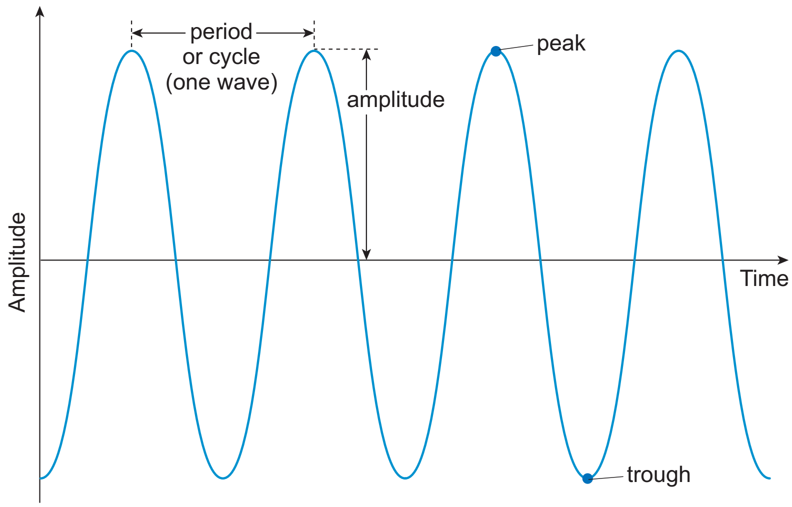


Figure 3.7 A waveform produced by a tuning fork

Of course, a tuning fork is a very simple instrument, and so makes a pure sound. Real instruments and real noises are much more complicated than this. An instrument like a clarinet would have a complex waveform, perhaps like the graph in Figure 3.8(a), and a dropped plate would be irregular like Figure 3.8(b).

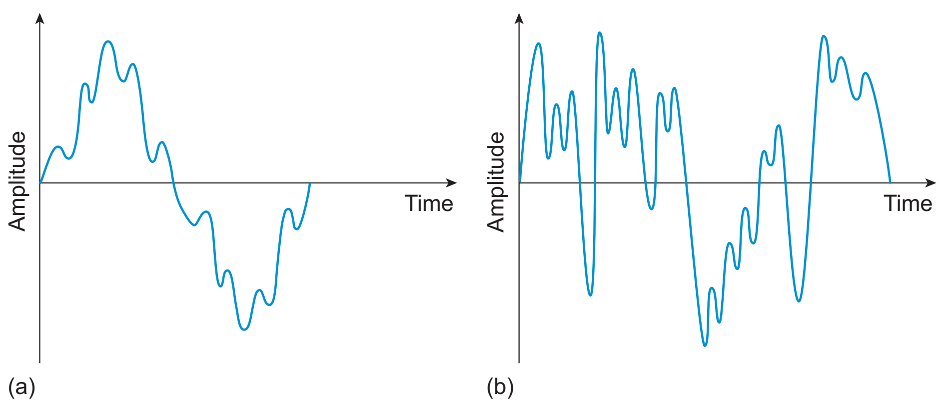


Figure 3.8 Typical waveforms: (a) a clarinet (b) a dropped plate

Any meaningful sound consists of cycles of pressure variations, i.e. the pressure variations vary in some sort of repeating pattern (as illustrated in Figure 3.9). These cycles may only last for a very short time, as in the case of a short sound such as a rifle shot, or they may last for a long time, as in the case of continuing sound such as a person singing a long note. Even for very short sounds, it is these cyclical pressure variations that allow us to differentiate the sound of, say, a rifle shot from a lightning crack; random variations of air pressure are perceived as noise. A regular pattern of high-and low-pressure regions like this is known as a pressure wave. Because the wave travels away from its source, it is also an example of a travelling wave, although travelling waves need not be pressure waves. Light, for instance, is a travelling wave but it is not a pressure wave. Among other things, a travelling wave is a way of transmitting energy.

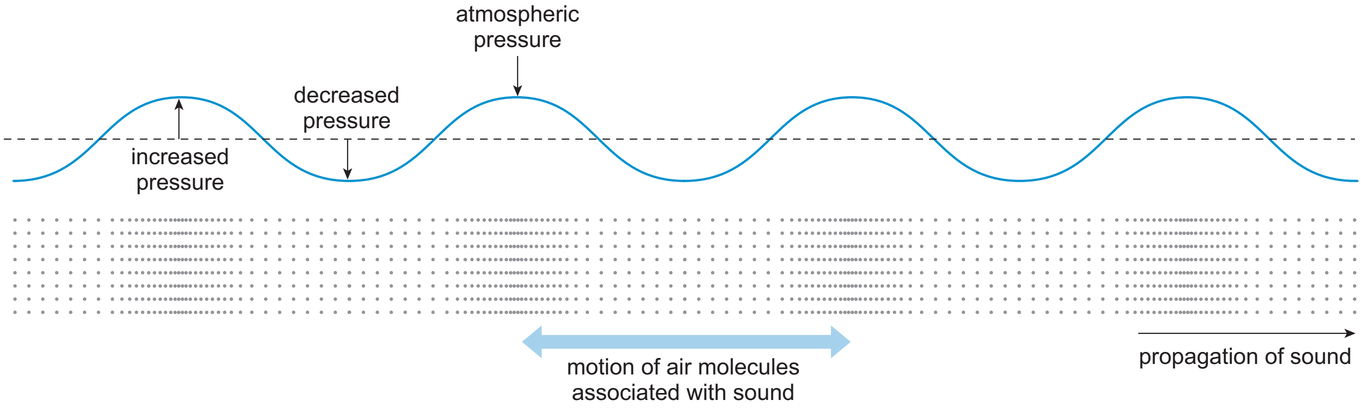


Figure 3.9 Sound waves as cycles of pressure variations

3.2.3 Sound sampling and storing

A microphone is used to convert the sound pressure wave into an analogue electrical signal with a voltage amplitude that varies in accordance with the pressure amplitude of the original signal. This electrical signal needs to be converted (coded) into a digital representation.

In the past, sound signals from microphones and other sound sources were mixed, stored and generally processed using only analogue methods. However, nowadays all these processes are almost always done using digital techniques.

What are the advantages of working with sound in its digital form? These are similar to the advantages of using digital techniques in other areas – immunity from signal corruption through extensive processing or through transmission or storage; and the fact that the mixing and processing of sound comes down to a simple process of computation ('number crunching'), rather than involving complicated analogue electronic circuits and devices. In addition, computer storage techniques can easily be used for storing sound in its digital form.

Digital sound

Working with sound in its digital form, often referred to as working in the digital domain, basically means working with numbers. How do we get the analogue signals collected by a microphone into a number form?

Essentially, we have to find some way to split up the waveform. As we split up images by dividing them into pixels, we can split a sound wave up by dividing it into very small time intervals, commonly called samples.

This is the process of analogue-to-digital conversion, which in practice can involve a number of quite sophisticated techniques and processes. However, in most cases the basic stages are the same – sampling followed by quantisation – which I will now explain.

The first stage of the process is to measure the amplitude of the sound wave at regular time intervals. Taking readings like this is called **sampling**. The number of times per second we take a sample is called the **sampling rate**.

I'll take the tuning fork example, set an interval of, say, 0.5 seconds and look at the state of the wave every 0.5 seconds, as shown in Figure 3.10.

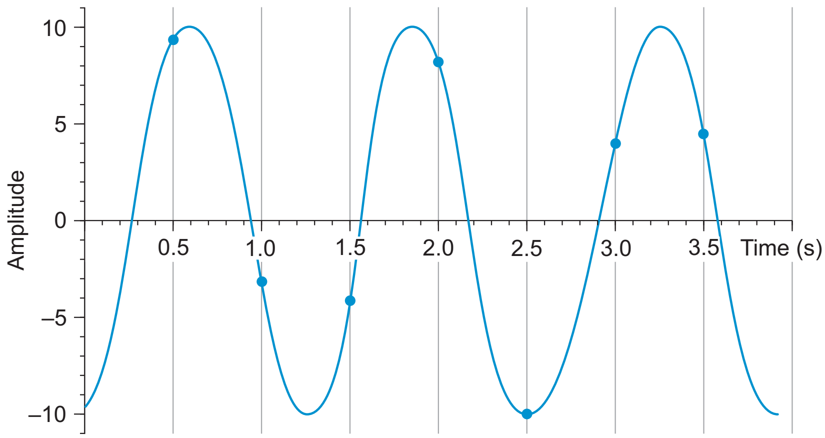


Figure 3.10 Sampling a sound wave at intervals of 0.5 seconds; dots represent sampling points

Reading off the amplitude of the wave at every sampling point gives the following set of numbers:

+9.3, −3.1, −4.1, +8.2, −10.0, +4.0, +4.5

as far as I can judge. Now, if we plot a new graph of the waveform, using just these figures, we get the graph in Figure 3.11.

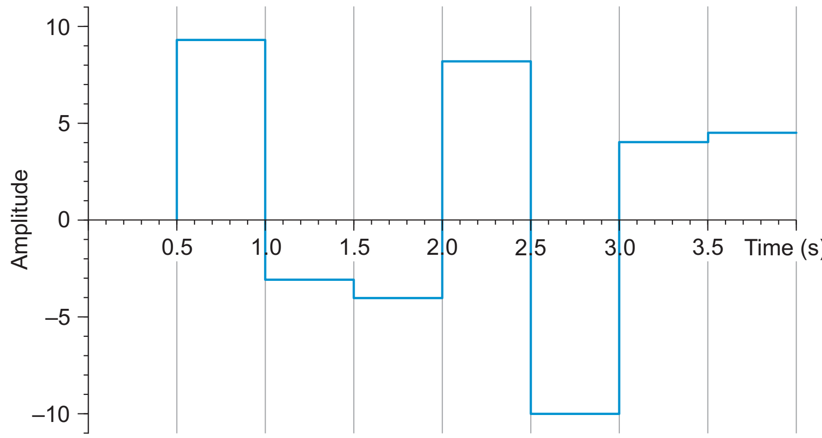


Figure 3.11 The digital conversion of Figure 3.10, with a sampling rate of 0.5 seconds

The plateaux at every sampling point represent the intervals between samples, where we have no information, and so assume that nothing happens. This means that the audio signal produced by the sample does not closely follow the original signal, leading to poor-quality sound.

Activity 3.11 (exploratory)

How can we improve on the amount of information shown in Figure 3.11?

Comment

We can make our sampling interval smaller – that is, we can decrease the temporal splitting up of the waveform.

So, let's decrease the sampling interval by taking a reading of the amplitude every 0.1 seconds, as shown in Figure 3.12.

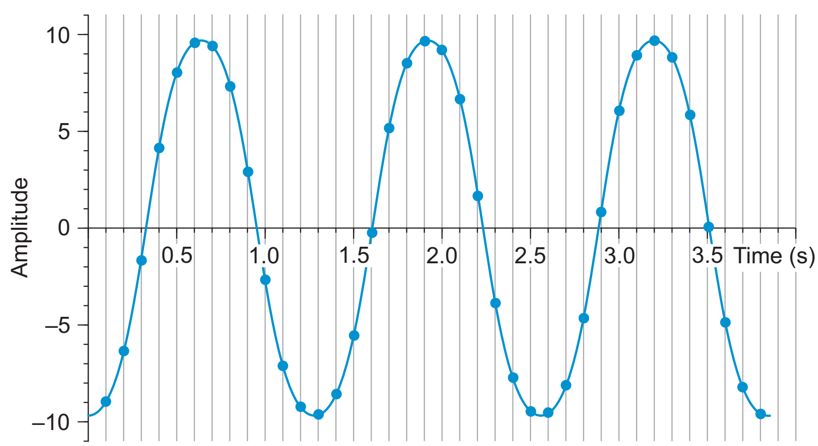


Figure 3.12 Sampling a sound wave at intervals of 0.1 seconds; dots represent sampling points

Once again, I'll read the amplitude at each sampling point and plot it to a new graph, as in Figure 3.13, which looks more similar to the original waveform.

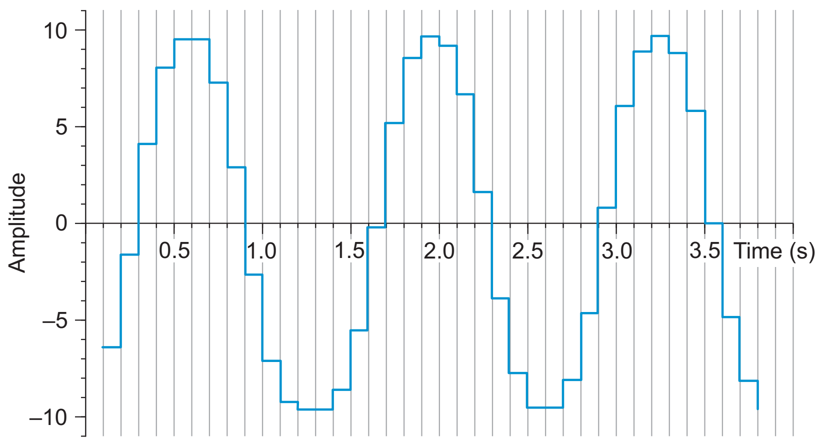


Figure 3.13 The digital conversion of Figure 3.12, with a sampling rate of 0.1 seconds

I hope you can see that this process of sampling the waveform has been very similar to the breaking up of a picture into pixels, except that, whereas we split the picture into tiny units of area, we are now breaking the waveform into units of time. In the case of the picture, making our pixels smaller increased the quality of the result; so making the time intervals at which we sample the waveform smaller will bring our encoding closer to the original sound. And just as it is impossible to make a perfect digital coding of an analogue picture, because we will always lose information between the pixels, so we will always lose information between the times we sample a waveform. We can never make a perfect digital representation of an analogue quantity.

The second step in the process – the quantisation stage – is to divide the maximum voltage range of the analogue sound signal into a number of discrete voltage bands. Each band is now represented by a number. Each sample is then mapped onto a band, and is therefore given the number which represents this band. The result is a string of numbers at regular intervals, as shown in Figure 3.14. Sound wave samples are generally mapped to 16-bit numbers representing the quantisation levels.

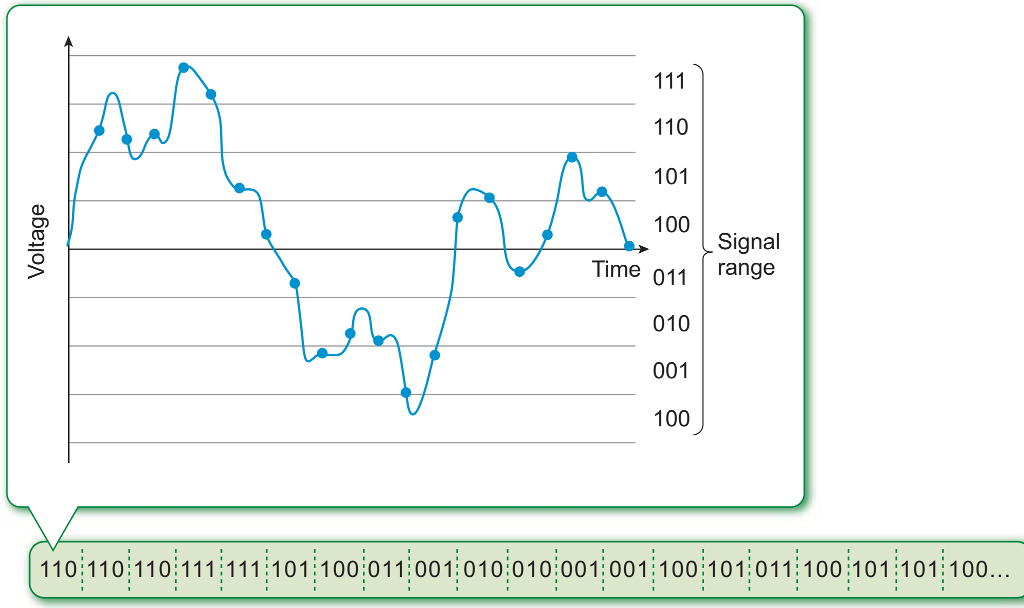


Figure 3.14 Quantisation

This mapping of samples to numbers is known as **quantisation**. Again, the faithfulness of the digital copy to the analogue original will depend on how large a range of numbers we make available. The human eye is an immensely discriminating instrument; the ear is less so. We are not generally able to detect pitch differences of less than a few hertz (1 hertz (Hz) is a frequency of one cycle per second). So the quantisation of sound usually involves a smaller range of numbers (and therefore less detail) than the quantisation of images.

3.2.4 Storing sound

Once a sound has been digitised, then the question is what do we do with the thousands of numbers that the analogue-to-digital converter outputs?

If we don't want to store the sound but just send it somewhere else, then we can simply send the string of numbers along a suitable transmission channel to a **digital-to-analogue** converter and loudspeaker at the other end. However, it is quite possible to send the string of numbers to a computer, and if the computer has enough storage, then all the numbers can be saved there.

Indeed, any device that can store large quantities of numbers can be used to store sound; such as modern music players, your phone's internal memory and any devices that contain a standard hard drive.

It must be remembered, though, that digital sound samples in their raw form do occupy a substantial amount of memory.

Activity 3.12 (practical)

1 Run Audacity and load the sound file associated with this activity, `Activity_3-12.wav`, which is a duplicate of the sound file you originally used in Activity 3.10.

2 Select the whole sound (`Edit > Select > All`), and determine the duration of the sound in seconds. (It is displayed in the end/length box in the `Selection` toolbar at the bottom of the Audacity project window.)

3 Using the sampling rate of the sound (also shown as the `Project rate` at the bottom of the Audacity project window), work out the number of samples the sound contains.

4 Now change the format so that the duration is given in samples, rather than time. Do this using the `Selection Start` drop-down list in the `Selection` toolbar. Does the number of samples given at the bottom agree with your calculation from Step 2?

5 Close the Audacity project using `File > Close` without saving it and in your computer's file system locate the file you opened in Step 1. Right-click on the file and select `Properties` (for a Mac, select the file and select `File > Get Info`), and note the size of the file in terms of the number of bytes it contains. (In Windows, this is the `Size:` figure, not the `Size on disk:` or `Compressed` figure.)

6 Suppose that each sound sample requires 2 bytes of storage, and that an additional 44 bytes are needed for format information about the sound (e.g. sampling rate, stereo/mono, duration). Is the size of the file what you would expect from your previous measurements?

Comment

The sound in the file has been adjusted to be exactly 4 seconds long. The sampling rate is 44 100 samples per second, thus the number of samples is four times this, or 176 400 samples. This should be the number that Audacity gives in Step 4. If each sample requires 2 bytes of storage, and an additional 44 bytes are needed for format information, then the file size should be $(176\,400 \times 2) + 44 = 352\,844$ bytes.

This is the storage requirement for just 4 seconds of single-channel (mono) sound. If the sound was in stereo, then the size would be doubled (ignoring the additional 44 bytes for format information).

3.2.5 Conclusion

In this section, I've described what sound is and have discussed how it is represented digitally. You have carried out some practical activities to explore the implications of using different sampling rates in the storage of sound files. In the next section, I'll be talking about sound recording, and you will be carrying out some practical exercises on the editing of sound.

3.3 Recording sound

Now that we have looked briefly at how sound can be digitised and stored, we can concentrate on looking at what we can do with the digital representations of sound, knowing that at the end we can use a digital-to-analogue converter and a loudspeaker to convert them back to sound again.

One of the main things we need to do with sound signals is to store them so that they can be played back at a later date. So I will start by looking at ways in which this can be done.

3.3.1 Making a recording

To start with, I'd like you to make a simple sound recording using Audacity. This will also give you the chance to look at the waveform of a typical sound signal.

Activity 3.13 (practical)

1 First make sure your microphone is connected to your computer, then run Audacity to open a blank project window.

2 Beside the microphone symbol there is a scale with the label '`Click to start monitoring`'. Click anywhere on the scale to activate the monitoring meter.

3 Say a few words into the microphone and see if the top (`L`) input level bar shows that Audacity is receiving sound from your microphone. If there is no movement in the microphone level bar, then you may need to adjust your input settings. Go to `Preferences > Devices` under the `Edit` menu (Windows) or the `Audacity` menu (Mac) and make sure the correct recording device is selected. In the `Recording` section, try choosing a different device from the one already selected (if there is one). Click on `OK` and see if the microphone bar does now respond to the sounds you make. If there is still no sound, you may need to go to your operating system's sound control panel and see if you can select the correct input there. Assuming your microphone is connected properly and is working, you should eventually be able to see that Audacity is responding to your voice.

4 Press the record button, a new track should appear and recording should commence. Say the following rhyme while recording:

Mary had a little lamb

whose fleece was white as snow.

And everywhere that Mary went

the lamb was sure to go.

I've used this rhyme for your recording as it was the one used by Thomas Edison in 1877 for the first ever sound recording to demonstrate his phonograph invention.

5 Click the stop button to stop recording.

Audacity gives you the option to save files as Audacity files but these are not intended to be read by other applications. Therefore, when you are ready to save a recording, you should save it as a WAV sound file by using `File > Export Audio`.You should give your file a suitable name such as `Activity_3-13.wav` or `Mary.wav`. Use a suitable location to store your sound file so you can easily locate it when you next need to access it.

6 Look at the waveform of the sound. Figure 3.15 shows the waveform of my effort at recording the rhyme. What do you notice about the overall shape of the waveform?

7 Play the sound through and watch where the cursor is for each word. Now zoom in on a section of the first word of the rhyme so that you can see the individual cycles of the waveform. What do you notice about the waveform now?

8 When you have finished looking at the waveform, close the Audacity project. There is no need to save the project, but do make sure you have saved your recording as a WAV sound file, as mentioned in Step 5 above.

Comment

In Step 6, you should have noticed that there are a few sharp peaks in the height of the waveform – which, if you play the recording, you will find correspond to consonants such as 's' and 'g'.

In Step 7, you should have noticed that, over a few cycles, the waveform can show some form of repetitive nature which indicates some sort of pitched sound; however, the shape of the waveform in each cycle is constantly changing.

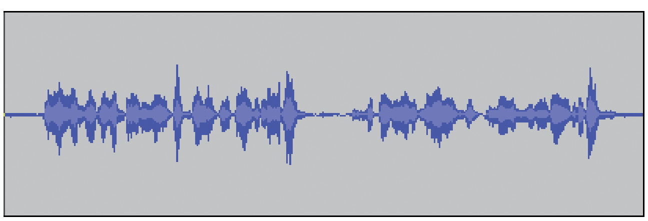


Figure 3.15 Waveform of a recording of 'Mary had a little lamb'

3.3.2 Processing sound

As long as the sound stays in a digital form, any sort of processing of the sound is simply a matter of 'number crunching' – and computers are very good at doing this. For example, if you wish to reduce the level of the sound by one half, then you simply divide each sample value by two. If you want to cut off the end of a sound, or perhaps move a section of sound to another place, then you simply delete or cut and paste the affected samples, respectively. Perhaps surprisingly, many standard effects such as the bass and treble controls on a hi-fi amplifier and even **reverberation** (the effect of multiple echoes that often occurs in a large church or cathedral) can be achieved by sophisticated number crunching of the sound samples.

When the processing of a sound is finished, the resulting modified sound samples can simply be fed at the correct rate to a digital-to-analogue converter and loudspeaker so that the result can be heard.

Editing sound

Although the term editing can cover just about any type of processing of digital sound data, basic editing usually means operations such as deleting or inserting sections of sound. Most other 'editing' operations usually have their own terms, e.g. fading, amplifying and adding effects.

Before we go too far into editing and the processing of sound in general, one point that must be considered is that of real-time operation. What I mean by operating in real time is that any processing of individual samples must be achieved within the sample period so that the sound can be played as it is being processed. If this is not possible, then all the processing will have to be done separately in advance, and the results stored before being played. Most modern computers are able to work sufficiently fast to be able to work in real time, but for some complicated processing you may find the computer has to complete all the processing before the resulting sound can be played. The same is true for any storage system – if the system cannot read the sound data out at the required data rate (or higher), then it cannot be used in real time. This doesn't mean that it can't be used to store sound, only that the sound must be transferred to a faster temporary store first before it can be played.

If the required processing of a sound can be achieved in real time, then there is the opportunity of carrying out what is called non-destructive editing. What this means is that all the editing operations are carried out on the fly using only the original sound data (which may involve many different sources of sound). The resulting sound samples are output directly to the digital-to-analogue converter and loudspeaker and are not stored, and so don't affect the original sound data, which remains intact. The advantage of this mode of working is that it is very easy to backtrack, whereas with destructive editing, where the original data is (or may be) overwritten with the processed data, this is not so easy to achieve – particularly where there are many sources of sound, and editing has been carried out incrementally in a number of stages.

In Activity 3.14 you will use Audacity to carry out some simple editing operations on a sound.

Activity 3.14 (practical)

1 Run Audacity and open the first file associated with this activity, `Activity_3-14a.wav`. Play the sound to remind yourself of what the speaker is saying.

Your task in this activity is to edit the sound such that the speaker says 'In my garden I have a *hazel* tree, an *apple* tree and a pine tree' rather than 'In my garden I have an *apple* tree, a *hazel* tree and a pine tree'. Remember that this activity is simply to give you some experience with editing a sound file, so don't spend too long in trying to achieve a perfect result.

2 Choose the `Selection` tool (the one that looks like the letter 'I' in the `Tools` toolbar). As accurately as you can, select the portion of the sound containing the words 'a hazel tree'.

Use the mouse to position the cursor and then hold the `Shift` key and click at the end of the selection. Alternatively, you can use `Shift` + the right-arrow key to select the section. If you hold down the shift key and position the mouse pointer over the selection, you will see the pointer change to a hand, which enables you to adjust the start and the end of the selection. You may need to use the zoom controls to zoom in on the section containing the required speech. Using the keyboard shortcuts here can help considerably: `Ctrl+1` for zooming in, `Ctrl+2` for zooming to a normal display and `Ctrl+3` for zooming out. (Use the `Command` key instead of the `Control` key with the Mac version.) Also, you may find it helpful to slow down the recording using the `Transcription` toolbar, which is the slider bar above the timeline on the right-hand side. Clicking on the green arrow will return the track to normal speed.

3 In order to make edits as 'clean' as possible and to reduce the possibility of introducing clicks at the edit points, it is always best to make edits at points where the waveform crosses the central zero amplitude point. Fortunately, Audacity can help you to do this. With your section of speech selected, select `Edit > Find Zero Crossings`. This will adjust the start and end points of your selection to the nearest sample with a zero amplitude. Indeed, to help to reduce further the likelihood of clicks, Audacity always selects the nearest zero crossing where the amplitude is moving from a negative to a positive direction.

4 Use `Edit > Cut` to remove the selected section of sound and copy it to the pasteboard (clipboard).

5 As accurately as you can, place the cursor at the point in the sound between the words 'have' and 'an' in the phrase 'In my garden I have an…'. Again, you will probably need to use the zoom controls to 'home in' on the best point. As before, you should also try to select a point that has an amplitude close to zero and is on the part of a cycle where the amplitude is moving from a negative to a positive direction.

6 Select `Edit > Paste` to insert the section of sound you removed in Step 4, and play the whole speech. How does it sound?

7 If you are very unhappy with the result, you can use Audacity's undo facility in the `Edit` menu to backtrack and have another try. However, as

mentioned in Step 1, do not spend too long on trying to get the result to sound perfect.

8 When you feel you have got a reasonable result, you can export the sound as a WAV file if you want to. Otherwise, you can just close the Audacity project without saving. My rough-and-ready attempt at the editing operation is supplied in the second file associated with this activity, `Activity_3-14b.wav`, if you want to compare your result with mine.

Comment

You should have been able to obtain a reasonable result without spending too much time. However, you will probably have found it quite difficult to get the selection and the insertion points right so that the edited speech still has a natural 'flow'. The other point to notice is that the speaker's inflection on the word 'a' in 'a hazel tree' sounds wrong when the section of speech is moved. Of course, there is nothing you can do about this.

Echo and reverberation

When the distance between a sound source and a surface is sufficiently large, the sound that is reflected is perceptibly delayed in comparison with the original sound, thus creating echo, the perceived repetition of a sound. Reverberation also results from sound bouncing off a surface, but repeatedly and in a way that means it's not possible to differentiate between the original sound and its repetitions. In this case, slightly delayed versions of a sound are added to the original sound, combining to create something entirely different. Reverberation adds 'body' to the sound.

In processing terms, echo is achieved by adding a slightly delayed version of the sound to the original sound. Two or more delays at different time intervals may be incorporated, and they are typically in the region of fractions of a second. Reverberation requires multiple delays to be added to the original sound in proportions that reduce as the delays get longer. In fact, some systems provide standard reverberation routines that reproduce the response of real rooms (e.g. a concert hall). Reverberation is also available on Audacity and is an excellent effect to apply to sounds that appear 'thin', to make them into an alternative that might have been recorded in a different space.

Activity 3.15 (practical)

Now that you've learnt how to do a simple recording, let's see how to add some effects. In this activity, I want to attempt to create something that gives the illusion of spatial movement (creating the echo phenomenon that you may have experienced in the 'real' world). This sort of effect is often best heard through stereo headphones.

These are the processes:

- apply the echo effect to the recording

- transform the original mono track into a stereo track containing a mix of the sound you have just created in the left channel and a delayed version of that in the right channel.

Here are the steps. (Remember that you can always undo processes if you don't like the results!)

1 Run Audacity and open the first file associated with this activity, `Activity_3-15a.wav`. Play the sound to hear what the speaker is saying.

2 Select the `Selection` tool from the `Tools` toolbar.

3 Select the whole of your recording. If you can't see the whole of the waveform in your window, go to `View > Fit in Window`. This will give you a complete view of your sound, and you can then select the whole of it.

4 Select the menu options `Effect > Echo`. This will open a small window in which you can enter the two parameters that control the transformation. The echo effect creates delayed copies of the original sound and adds these copies together to create a new sample. For my own recording, I used 0.5 seconds for `Delay time` (which controls the delay time between the various copies) and 0.4 for `Decay factor` (which controls how quickly the copies 'disappear'), but you may want to experiment a bit on your own recording.

5 Select the menu options `Tracks > Add New > Mono Track`. This will create a new audio track right below the one you've already got. Click to reveal the drop-down menu on the Audio Track Control Panel of the new track and select `Right Channel`. Then click on the drop-down menu of the original track and select `Left Channel`.

6 Using the `Selection` tool, select the whole of the sound you've now got in the left channel. Copy that to the clipboard by choosing the `Edit > Copy` menu options or the `Copy` button from the `Edit` toolbar.

7 Click anywhere on the right channel track and paste the sound there using `Edit > Paste` or the `Paste` button from the `Edit` toolbar.

8 You should now have exactly the same sound in both channels, but to make the result more interesting, you're going to add a delay to the right channel.

9 Select the `Time Shift` tool from the `Tools` toolbar (it looks like a double-headed arrow). You'll see that this changes your cursor accordingly. Click anywhere on the waveform in the right channel and drag the whole of it to the right. I dragged mine to add a delay of about 0.3 seconds, but again you may want to experiment with your sound to achieve a result that you like.

10 To join the two separate mono tracks into a single stereo track, use the selection tool to select both tracks, then click on `Tracks > Mix and Render`. Remember to use `View > Fit in Window` if you want to see the whole of the resulting waveform. In this case, you should see two waves: one for the right channel and the other for the left channel.

11 You can then save your project and/or export as an audio file. You may want to experiment with other sound effects, such as `Amplify`, `Fade In`, `Fade Out`, or `Invert` .

Comment

You'll find my own result on the module website. The file is called `Activity_3-15b.wav`. The actual process was fairly simple, yet I was quite pleased that I'd achieved the original purpose of creating the illusion of spatial movement. I think the resulting sound is much more interesting than what I had initially (though, of course, you're free to disagree!).

3.3.3 Mixing sound: TMA preparation

In the next activity, I want you to use Audacity to mix together a number of different sound sources. This is something you may well have to do when you are creating the soundtrack for your TMA.

Activity 3.16 (practical)

1 Run Audacity and open the first sound file associated with this activity, called `Activity_3-16a.wav`. Play the file and you should hear some music which you will be using as background music.

2 Use `Project` > `Import` > `Audio`… to add an additional soundtrack to the Audacity project. Load the voice recording in `Activity_3.16b.wav`.

3 Use `File` > `Save Project As`… to save the Audacity project.

Your task now is to adjust the background music so that while the speaker is talking the volume of the music is reduced. Finally, a short while after the end of the speech, the background music slowly fades away completely.

4 Select the `Time Shift` tool and use it to drag the voice soundtrack horizontally to the right so that it starts about 9 seconds from the start of the background music. (The precise point is not critical.)

5 Play the sound and check that you can hear the background music playing, and that the speech can be heard, on top of the music. Save the Audacity project.

6 Select the `Envelope` tool (the one with two triangles and a thin blue line) which is directly above the `Time Shift` tool). A wide blue border, the 'envelope', will appear around the waveform. This tool allows you to change the volume envelope of a track as the sound plays. You do this by creating 'handles' that you can drag up and down to vary the volume, and left and right to alter the point at which the volume change starts.

7 About 1 second before the speech track starts, click in the background music track (either of the stereo channels) to create a handle which looks like four small circles. Click about 1 second after the start of the voice track to create a second handle. Now drag the first handle upwards so that the size of the first section of the music is as it was before the handles

were created (i.e. the music is at full volume). Now drag the second handle so that the track height is about one quarter of the full height. (Drag a handle out of the track area to delete it if you accidentally create more than two handles.)

8 Create another handle a few seconds before the end of the rhyme. Adjust the 'levels' of the handle such that it is at the same height as the lower of the first handle.

9 Save the Audacity project. The resulting Audacity screen should look something like the one shown in Figure 3.16.

10 Play the complete sound and check that the background music does indeed fade during the speech, and the music does then fade out completely at the end.

11 If the music levels are not quite right or fade at the wrong points, you may like to adjust the amount and/or positions of the fades to get a better result, but do not spend long on this.

12 When you are happy with your final result, save the Audacity project and then close it.

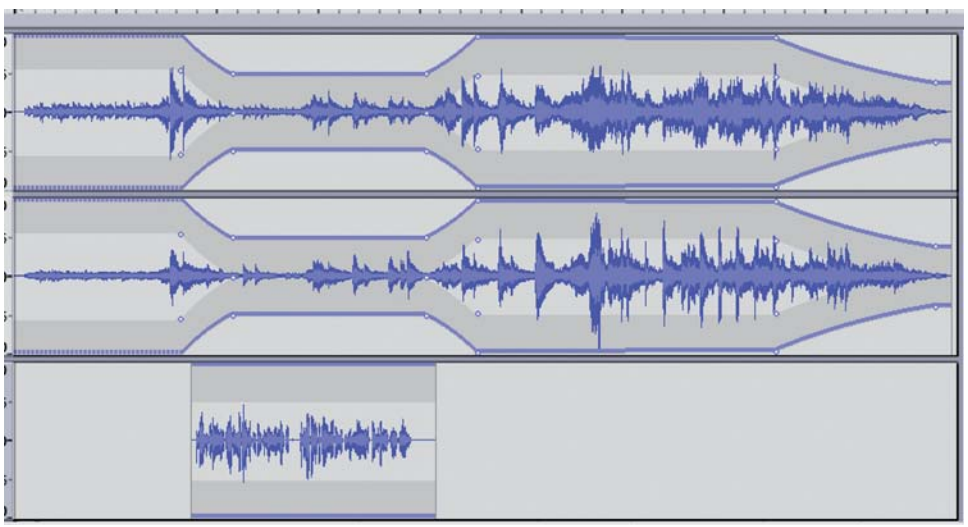


Figure 3.16 MIX Audacity project after incorporation of envelope and fading operations

Comment

Although you won't need to use this Audacity project again, you may find it useful to have it saved so you can look back on what you did when you come to create the soundtrack for your own audio file in the TMA.

Notice how Audacity allows you to drag sounds so that they can be made to start at different points and not necessarily always at the start of the track.

What the envelope tool is doing is essentially allowing you to create an edit list that contains instructions on when and how to adjust the music volume when the sounds are playing.

This activity has shown you how to mix a number of different sound sources together; how to alter the points at which the various sounds begin; and has shown you how to create dynamic volume changes as a sound is playing. These are all skills that you are likely to need to do when creating your own audio file in the TMA.

3.3.4 Conclusion

In this section, I have introduced how to make recordings using your computer and its microphone, how to view the sound waveforms and edit the soundtrack as well as how to add simple sound effects and mix together a number of soundtracks. In the next section, I'll be talking about compression of digital files including sound and images.

Compression of digital files

3.4

The digital encoding of sound, images and, especially, video produces large data files. As an example, the average file size of a 500-page text document created in plain-text format in Microsoft Word is 1.8 MB, whereas a simple 30-second-long video may be 186 MB. Such large files create important constraints in that they require a lot of storage space, it is more laborious to transfer the data across storage media, and transmission across computer networks such as the internet takes longer.

Compression means taking the original digital file and converting it into a new file that uses considerably fewer bits without significantly degrading the quality of the original content. Thus, compression is a way to reduce the number of bits used in encoding the data, so we can save on the memory needed to store the data and also increase the speed of transfer. The ratio between the uncompressed file size and compressed file size is known as the compression ratio.

$$\text{compression ratio} = \frac{\text{size of the uncompressed file (original file)}}{\text{size of the compressed file}}$$

The higher the compression ratio is, the less storage space would be needed for the resulting compressed file. Therefore, if a 10 MB file is compressed to 2 MB, the compression ratio is 10:2, which is equivalent to 5:1.

In this section, we'll look at a few common techniques used for audio and video compression. There are two main types of compression, **lossy** and **lossless**, and I'll now look at each in turn.

3.4.1 Lossless compression

With lossless compression, the digital data is stored in a compressed form such that it can be recovered, sample for sample, with nothing altered. Nothing is taken away or lost – as the name implies. Such lossless data compression is commonly used by the computer industry to reduce the amount of storage needed for a particular item of data. A computer ZIP file (created using the popular Winzip compression software) is an example of lossless compression of digital data.

A powerful form of lossless coding is called **dictionary-based coding**. An example of such coding is the widely used 'Lempel–Zev–Welch', or LZW algorithm, found in the Winzip software.

In most digital files, some data is repeated over and over again. A compression algorithm simply gets rid of this redundancy by scanning for patterns, and translating those patterns into something that takes up less space. I'll use an example of compressing some text to demonstrate how this works.

When the going gets tough, the tough get going.

In the modern proverb shown above there are 9 words, 8 spaces and 2 punctuation symbols: a total of 47 characters. If one unit of memory is allocated for each character, the file will consist of 47 units.

On closer examination, you can see that some words occur more than once; ‘the’, ‘going’ and ‘tough’, are repeated. We can construct almost half of the second phrase from the first half of the sentence. Our ‘dictionary’ of words looks like this:

1 the

2 going

3 tough

The sentence can now be expressed as:

When 1 2 gets 3, 1 3 get 2.

Thus, the original sentence uses 47 units of memory while the compressed version uses only 27. If you know the code you can reconstruct the entire original sentence.

Another technique for lossless compression is called run-length encoding. In this compression method, ‘runs’ of similar data items are recognised and replaced with a single instance of the data item, together with a number indicating how many times the item is repeated. For example, the striped pattern in Figure 3.17 can be represented by the binary string 01010101010101. The string repeats the sequence 01 seven times. This can be represented in run-length encoding as 7(01).

Figure 3.17 A striped pattern

Lossless coding is used in a similar way for simple images, such as line drawings and simple shapes in the GIF (Graphics Interchange Format) format used for images on websites.

3.4.2 Lossy compression

An easy-to-understand example of lossy compression is the old-style telegrams, which were used to send messages over wire cables. Telegrams were very costly as companies would charge for their services by the number of words. In order to reduce the cost, people usually sent short messages, taking off some of the words and hoping that the person receiving the message would still be able to understand the meaning – hence sending a maximum amount of information using a minimum number of words. For example, a message like:

> ‘Hello, I will be celebrating my graduation in Cornwall Town Hall next Friday November 12. I hope you can come with your family. Can you please let me know if you can’t come?’ 33 words

would be compressed as

> ‘Graduation next Friday Cornwall Town Hall. Come family. Let me know.’ 11 words.

The message could be understood as the general meaning was not changed. This was possible because some of the words do not add much to the sense of the sentence or were simply redundant. Although it is possible to understand the compressed message, the 22 words taken off the original message would be lost and cannot be recovered – hence the name lossy compression. Similarly, audio, image and video files can be compressed by removing some of the information that does not add much to the file or is simply redundant.

Lossy compression of audio, image and video files exploits features of human perception to reduce the amount of data needed. I’ll now look at how lossy compression is used for each type of file.

3.4.3 Psychoacoustics and audio coding

Psychoacoustics is the study of how humans perceive and respond to sound. Knowledge from psychoacoustics is used in lossy compression techniques to reduce the amount of information a sound signal contains without affecting the perceived sound. This is achieved by exploiting two particular aspects of how human beings respond to sound.

Firstly, the sensitivity of the human ear varies according to the frequency of the sound. For example, one sound at a given frequency and amplitude may be heard, whereas a sound at a different frequency but with a higher amplitude cannot be heard. Thus it is possible to define so-called ‘hearing thresholds’. By detecting which parts of the signal are below the threshold and removing these components, the amount of information can be reduced.

Secondly, consider two sound sources emitting different sounds at the same time. If one of the sounds is much louder than the other, and if they are of a similar frequency, then the louder sound may totally mask the softer one. To illustrate this, consider two people having a conversation; if a noisy aircraft passes close overhead, it is quite likely that for a certain time the range of frequencies emitted by the aircraft’s engines will mask the conversation. Lossy audio compression can use this effect to remove the parts of a sound signal that are more than a certain level below a loud sound, and thus would not be heard. This again reduces the amount of information that the sound signal contains and hence reduces the number of bits required to store such a signal.

MP3 audio compression

MP3 audio compression offers acceptable audio quality with a high compression ratio in the region of 11:1 (read 'eleven to one'). An MP3 encoder operates as follows.

The audio signal is split into 32 frequency bands that match the frequency characteristics of the human ear. The sound content of each band is then analysed and coded using a specialised algorithm which uses the lowest possible amount of data for the given content by exploiting the psychoacoustic aspects explained above.

When an MP3-audio is to be played back, decoding requires the data blocks to be separated out so that the frequency data can be reconstructed and used to rebuild the original signal. Therefore, it should be perceived by the listener to be the same as the original signal.

There is an optional practical activity on compression and MP3 available on the module website.

3.4.4 Digital image compression

As described in Section 3.1, a digital image is made up of multiple small elements known as pixels. The more pixels there are per unit area of the image, the higher is its resolution, but the greater is the storage requirement in terms of memory space. There are two solutions to reduce the amount of memory required: either reduce the number of pixels required (therefore reducing image resolution) or compress the data using compression software. In the following, I will introduce the main compression formats.

One of the most important standards for photo image coding (also used as part of MPEG video) is the one produced by the Joint Photography Experts Group – **JPEG**. If you're a user of a digital camera, you'll be aware of JPEG files (file extension .jpg). This compression technique divides a picture up into small blocks of pixels and performs complex calculations to arrive at a reasonably accurate but concise description of the block. Just as we can remove or reduce some frequencies in an audio file because they're not important for human hearing, we can process images in a similar way and remove or reduce some components because they're not important for human vision. An interesting point about JPEG is that it is a lossy compression method and the original data can never be recovered exactly – only an approximation to the original can be recovered. Since the human eye simply does not detect some degradation in images, we are unaware of the effects of the compression process. Files in JPEG format can often be a small fraction of the directly encoded bitmap file without appreciable loss of quality as perceived by most people. JPEG can achieve compression ratios of 10 to 20 with no visible loss of quality and ratios of 30 to 50 if some loss of quality is acceptable.

To a certain extent, quality depends on the type of image you're dealing with as well as the uses to which the image is put. Whilst you'll be working mostly with JPEG files on TM111, other common image formats available on the web include GIF (Graphics Interchange Format) and PNG (Portable Network Graphics). Both of these use lossless compression methods.

Figure 3.18 shows you the effects of compression on an image. You should notice that increasing the compression ratio creates images of lower quality.

(a)

(b)

Figure 3.18 The effects of compression on an image: (a) low compression; (b) high compression

Activity 3.17 (exploratory)

Here are some facts about image recording:

- Human perception is less sensitive to colour than it is to brightness.
- Information about colour and brightness can be coded and recorded separately.

Can you suggest how you might exploit this information in the coding of colour images so as to reduce the file sizes?

Comment

If human perception is less sensitive to colour than it is to light intensity, then it makes sense to use fewer bits to code colour information than are used to code the brightness.

3.4.5 Conclusion

In this section, I have introduced you to techniques used to compress audio and image and data files. We've also touched on how techniques from the study of human perception are used in compression.

3.5 Developing multimedia content for your TMA

This section relates to your work for TMA 01 and is on the module website. You will also find details of the assignment on the module website.

Summary

In this part, you have learned about how popular media such as images and sound can be converted into digital forms.

You have learned more about the representation of data in binary form, e.g. how bit words can represent colour. You also learned about:

- file formats used for images
- how compression is used to reduce the size of data files so that they take up less space in memory and can be transmitted across networks more quickly
- the different types of compression: lossy and lossless.

It has only been possible to give you a very brief outline of the physics/ technology behind sound and its recording and processing in this part of Block 1. However, I hope that you now have a good grasp of the basics of what sound is, and how it can be recorded, processed and stored when in a digital form.

Importantly, this introduction has also provided you with some valuable skills with the Audacity sound editor program, and you will use these when creating your audio file for your TMA. In particular, you should now be able to use Audacity to:

- load and play sound files
- make recordings using your computer and its microphone
- view sound waveforms
- edit sections of sound
- add sound effects
- mix together a number of sound sources.

Finally, the skills you developed in producing your audio file are important transferrable skills that are widely used in the workplace.

Answers to self-assessment activities

Activity 3.2

I'd say that (d) and (e) are fundamentally discrete. Examples (a), (b), (c) and (f) vary continuously.

Activity 3.3

An 'analogue' representation is a direct counterpart of a physical quantity. A sound signal, for example, can be converted into an electrical 'analogue' and then stored or transmitted. The variations of the electrical 'analogue' show the same pattern of variations as those in the original sound wave. Analogue representations can generally take on any value within a range.

In a 'digital' representation, a limited set of values is used to store, set or display a coded version of the original. A digital representation can only take on certain values. Examples of digital representation include a digital photograph and the numerical display of a digital thermometer.

Activity 3.5

Change all the 0s to 1s, and all the 1s to 0s.

Activity 3.6

(a) For the higher-resolution display, we have a total of $2560 \times 1600 = 4\ 096\ 000$ pixels.

For the lower-resolution, we have $1024 \times 640 = 655\ 360$ pixels.

(b) Therefore, the ratio of the two resolutions is 1:6.25.

Activity 3.7

We can generate the new words in the same way as before. First, take the previous eight 3-bit words, and add a binary 0 to the front. Then do the same with a binary 1. Now we have sixteen 4-bit code words for sixteen different colours, as shown below. (I haven't tried to think of sixteen different colours!)

You may have written out the binary codes in a different order, but the order given is the standard way of doing this, and includes all possible codes for the 4-bit words.

0000	1000
0001	1001
0010	1010
0011	1011
0100	1100
0101	1101
0110	1110
0111	1111

Activity 3.8

(a) In this example the data needed for each image is calculated as follows:

The display is 1024 × 768 pixels = 786 432 pixels.

Each pixel uses 24 bits, so the total number of bits required is: 786 432 × 24 = 18 874 368 bits.

(b) The upload speed is 2.3 Mbps, so we convert this value to bits: 2.3 × 1000 × 1000 = 2 300 000 bits.

Using time = size/speed:

18 874 368 / 2 300 000 = 8.206 246 956 521 74 = 8.21 seconds, to two decimal places.

References

Alien (1979) Directed by Ridley Scott [Film]. London, 20th Century Fox.

McLuhan, M. (1964) *Understanding Media: The Extensions of Man*, New York, McGraw Hill.

Acknowledgements

Grateful acknowledgement is made to the following sources.

Figure 3.1a. © Stbusypix/www.istockphoto.com

Figure 3.1b. © Tom Hahn/www.istock.com

Every effort has been made to contact copyright holders. If any have been inadvertently overlooked, the publishers will be pleased to make the necessary arrangements at the first opportunity.

Part 4 A world built of data

by Mike Richards

Contents

Introduction

Can I afford those new shoes? Which school should we send our child to? Is the design of this airliner safe? Is climate change driven by human activity? We can answer these questions by processing data. Some questions we can solve in our heads or on a piece of paper, but, increasingly, the answers we need as individuals and as societies can only be obtained with the help of computers.

Storing and processing data inside computers has changed us as individuals and as a society. Whether or not you realise it, or indeed want it, governments, companies and other computer users use data associated with you to determine everything from your income, your health, your likelihood to commit crimes or whether or not they want to be your friend. Some of this data you create consciously; such as when you post on social media. Other data is created whenever you open a bank account, enrol on a module, shop using a store loyalty card, and so on. And yet more data is created without your knowledge by companies and governments who monitor your interactions with computers – when you make a phone call and to whom; your web browsing and email histories; your financial transactions and your journeys around the world.

Activity 4.1

Take a moment to look in your wallet or purse to see what cards and documents it holds. Each of these is likely to mean that some organisation holds electronic records about you. What kind of information do you think these organisations hold about you? (Bank notes and coins do not hold any personal information so do not include them in your list.)

Comment

My wallet contains: my office pass, a bank card, two credit cards, six store loyalty cards (clearly I'm not a very loyal shopper!), a museum membership card, two airline loyalty cards, a newspaper membership card, a political membership card and a pre-paid coffee shop card.

Your wallet or purse may contain a similar range of items. The point is that various organisations (including my bank, favourite shops and airlines, museum and coffee shop) all hold some data about me. They probably hold my name, age, date of birth and address, plus some data unique to each particular organisation. For instance, an airline knows which flights I have taken with them – but the coffee shop does not. The coffee chain can record which drinks I order using my card, but the airline does not know that I prefer skimmed milk in coffee. Clearly, different organisations each have very different views of me.

This part explores the relationship between data, computers and humans. It will give you a broad overview of how computers manage data and how that data can be made useful. I will start with a brief exploration of what we mean by 'data' and then introduce one of the most powerful tools for understanding data ever created: the database.

Databases made it possible to query huge collections of data to find individual pieces of data or uncover subtle patterns within the data at very high speed. They have revolutionised almost every part of our life, from travel and shopping to identifying criminals, providing better healthcare and supporting scientific investigations. Databases underpin most businesses as well as a large part of the internet. I will explain how databases are constructed and used, and you will have a chance to interact with a simple database to develop your understanding of the topic.

This part ends by introducing the next data revolution: so-called big data. Modern technology now allows us to collect and process an almost unimaginable amount of data, ranging from the latest scientific investigations into space, physics and genetics through to everyday tasks such as identifying traffic flows in cities and helping us find useful websites. This volume of data requires new approaches, so we will explore exactly how computer scientists are using big data to help scientists, engineers, healthcare specialists, businesses – and the rest of us – solve a new generation of problems.

What is data?

4.1

Data can assume any form – numbers, characters, pieces of text, pictures or sounds – but crucially data has no context. So here is a piece of data:

> 8848 metres.

This lack of context means that it is very hard to work out the meaning of that data; 8848 metres is equally valid as:

(a) the distance covered during a pleasant afternoon walk, or

(b) the height of Mount Everest,

which we can agree are two very different things! Data, on its own, isn't very useful; instead, data is turned into useful information by placing it within a context, such as:

> According to my Fitbit, we have just walked 8848 metres.

or:

> Towering 8848 metres above sea level, Mount Everest is the tallest mountain on Earth.

Computer programs process data; but, crucially, the computer has no understanding of whether it is calculating the length of a walk or the height of Everest when it receives the data 8848 metres. The computer simply follows the instructions held in the program. Computer programs transform data into information by imposing structure on data to make it meaningful to human beings.

At this point, I need to define what we mean by data. Here is a working definition that is suitable for this module:

> Data is one or more values that can be assigned to an object.

Let's explore this in a little more detail. **Values** are familiar from everyday life – names, prices and titles of books or movies are examples of values. **Objects** aren't just physical entities that we are used to in everyday life (such as a mountain, person, city, and so on), but can also be virtual, without a physical existence (e.g. a character in a novel or tomorrow's weather forecast).

Box 4.1 Is data singular or plural?

Traditionally data is always used as a plural noun, so I could write: ‘Data from thousands of patients *are* processed in the cloud’. However, computer scientists, as well as almost all journalists, tend to treat data as a singular noun, so it is now more common to write ‘Data from thousands of patients *is* processed in the cloud’. Either form is acceptable, but it is crucial to be consistent in your writing!

4.1.1 Organising data

As soon as we start to acquire data, we create a new problem of accessing that data with the minimal expenditure of time and effort. This problem was solved long ago for books. You can quickly decide if a book will be useful without even opening it. Information such as the title and author, and occasionally a summary, are printed on the cover. A quick glance will tell you if the book is relevant. However, finding a fact or your favourite section of that book would be a long, difficult task if there was no organisation within the book itself.

Books are divided into sections and chapters which group related content into chunks, again allowing you to quickly find relevant material. Navigation is made easier by a table of contents listing chapters and sections, whilst individual words and concepts can be located using an index or glossary.

Books themselves can be organised into related categories and subcategories in book shops and libraries to make them easy to find and compare to one another. Think of the shelving in a large store, which might include a series of increasingly specific categories, such as those below:

Fiction – Crime novels

- British crime
- Classic crime
- Historical crime
- Nordic crime
  - Danish crime novels
  - Icelandic crime novels

This idea of labelling, indexing and cataloguing information is not unique to books. Similar schemes were developed long before computers were even dreamt of; but they went by names such as censuses, registers of births and deaths, electoral rolls, manifests, company ledgers, and so on.

Today, when we want to store large amounts of data we use a **database**.

4.1.2 Databases

For the remainder of this part the discussion will be restricted to computer databases. We can define such a database as:

> A collection of related data organised to permit efficient access for reading and updating by a computer.

The first known example of a computer database dates from 1962, when the words ‘data base’ appeared in a technical memorandum by the world’s first software company, the System Development Corporation of California. Databases proved to be one of the crucial applications that drove the computerisation of big businesses and government during the 1960s. Two databases proved to be especially influential: SABRE and IMS.

SABRE

Until the early 1960s, buying an airline ticket was a lengthy and painstaking task. A traveller would have to visit a ticket agency, where a ticket agent would take down their requirements and then telephone a regional booking office to enquire about seat availability. At the booking office, a clerk would locate a piece of card representing an individual flight (see Figure 4.1). Each card contained boxes, one for each seat on the aircraft. If any of the boxes were empty, a seat was available and the clerk would then take down the traveller’s details to make the reservation itself. If the flight was full, the clerk would have to find another card and check if any seats were available on that flight.

In 1961, American Airlines alone made some 20 000 reservations per day, with each reservation taking an average of 90 minutes to complete. Naturally, with all records being stored using pen and paper, the system was bedevilled by mistakes and lost bookings.

Figure 4.1 A typical airline booking office from the late 1950s. Individual flights are stored in folders on the rotating circular shelves in the middle of the office

A chance encounter on an American Airlines flight between IBM salesman R. Blair Smith and C.R. Smith, the president of American Airlines, led to a contract between the two companies. IBM’s tortuously named Semi-Automatic Business Research Environment (SABRE) was the first civilian computer application to provide real-time responses to enquiries. Previously, computer users had to wait minutes or even hours for their answers; SABRE offered flight availability in seconds.

Flights were booked using teletype terminals like that in the foreground of Figure 4.2. An agent would make a reservation by placing a card listing flights to a destination on the upright part of the terminal and then pressing a 'NEED' button. The SABRE database would look for available seats on those flights. If a plane was fully booked, a light would come on next to the unavailable flight. Assuming a flight was available, the agent would then complete the booking using the teletype keyboard.

Figure 4.2 A publicity photograph for the first SABRE computer system

SABRE went live in 1961; some 1500 terminals, located at American Airlines' own offices and in ticket agencies across the country, used telephone connections to communicate with a pair of IBM 7090 computers located in New York State. By 1964, all of American Airlines' bookings were being made through SABRE at a rate of 7500 reservations per hour. SABRE was the largest private, real-time data processing system in the United States and gave American Airlines a huge competitive advantage over rival airlines. SABRE not only served customers more quickly and reliably, but gave American Airlines immediate access to figures such as passenger numbers on routes and the number of unsold seats on forthcoming flights. Such was SABRE's success that rival airlines either paid for access to SABRE or began development of their own competing systems.

With its rapid response to customer queries and use of networks, SABRE has a good claim to be the ancestor of all the e-commerce systems we use today, from online ticketing to shopping and banking. In 2000, American Airlines spun-off SABRE into a separate company – Sabre Holdings – which has continued to develop and expand the database into a worldwide service accessible from any internet-connected computer. Officially known as the Sabre Global Distribution System, SABRE is a $3.2-billion-per-year business serving over 400 airlines, 17 cruise ship companies and more than 100 000 hotels. An estimated 57 000 travel agencies use SABRE every day, performing more than 42 000 transactions every second. If you have travelled recently, it is more than likely SABRE helped you on your way.

IMS

Shortly after the introduction of SABRE, IBM, in collaboration with North American Aviation and the Caterpillar Tractor Company, began development of the Information Management System (IMS) database. IMS was needed to solve the greatest engineering problem of the 1960s: how to build the Saturn V rocket that would carry Apollo astronauts to the Moon. Standing 36 storeys tall, the Saturn V was one of the greatest pieces of twentieth-century engineering (see Figure 4.3); it was by far the largest and most sophisticated rocket then flown.

IMS kept track of the nearly 3 million components in each rocket and was responsible for scheduling delivery of components to meet NASA's deadlines, as well as ensuring contractors were constantly kept informed of changes to the design of the rocket or to the Apollo programme's schedules. Without IMS, it is highly likely American astronauts would not have landed on the Moon in 1969.

Figure 4.3 A view from the top of the Vehicle Assembly Building in late 1967 at the Kennedy Space Center, Florida, where the first Saturn V rocket was being prepared for the launch of Apollo 4. For scale, the blurred image of a person can be seen in the red gantry just to the left of the rocket's nose cone.

In 1968, IMS was released to IBM's corporate clients and proved to be an immediate success. Indeed, IMS remains a core part of IBM's business and a crucial part of the world economy. IMS is used by 90% of the world's 1000 largest companies in areas as diverse as defence, banking, aerospace and pharmaceuticals to process more than 50 billion transactions every day. Without knowing it, you've probably interacted with at least one IMS database today.

4.1.3 Databases and your view of the world

SABRE and IMS were just the first of a huge number of databases which quickly found their way into every part of our lives. The benefits of databases were immediate, profound and widely publicised, but one of their potentially harmful consequences was rather less publicised; the designers of a database could make data more or less visible simply by changing how data was retrieved.

Facing increased competition in the 1980s, American Airlines modified SABRE to ensure it always listed American Airlines' flights ahead of other airlines. Travel agents preferentially selected American Airlines even when other airlines were cheaper or faster because they perceived the first few results to be the 'best' for their customers. Eventually, the American government intervened to outlaw the practice which it regarded as anticompetitive and disadvantaging smaller airliners.

American Airlines' manipulation of SABRE's results to favour their flights showed how supposedly impartial databases could be used to favour certain data over others. We should bear this in mind whenever we interact with services powered by databases. We presume web search engines, price comparison websites, online travel agents and online stores rank results from best to worst, so we almost always choose one of the first options presented to us. But have those results been manipulated to drive us towards certain results?

Web search is the most competitive of all services, and everyone wants to be one of the first results returned. Studies monitoring the behaviour of search engine users show that 32% of people select the first search result and another 18% click the second link. The tenth result gets just 2% of all clicks and later results are even less popular.

With more than 95% of users only following links on the first page of results (van Deursen and van Dijk, 2009), entire businesses are devoted to improving the 'search engine ranking' of their clients. This process is known as **search engine optimisation (SEO)**.

Activity 4.2 (exploratory)

The Guardian is about the 150th most popular site on the entire world wide web and is in part supported by advertising revenue. As well as hosting adverts *The Guardian* site offers advice to advertisers in the article 'What is SEO and how can it help my website's Google visibility?' (Lines, 2014). Read through this article, provided on the module website, and take notes to answer the following three questions:

(a) Why is SEO important?

(b) How does the article recommend improving an organisation's search engine rankings?

(c) Why is keeping material up-to-date important?

Comment

(a) SEO is a method of increasing the ranking of a site in a list of search results, with the aim of improving the number of people visiting a link. For commercial advertisers, this could attract more customers to their business.

Although search engines try to judge the quality of the pages they link to, it is quite possible for pages with high-quality content to appear further down the search listings, either because they are not widely linked to from other sites or because they do not use keywords correctly. This fact is directly relevant to your studies; do not confine your research to following just the first two or three items in the results!

(b) A listing can be improved in two ways:

(i) by using keywords that the search engine recognises as relevant to the site's purpose

(ii) by linking to other high-value sites and social media accounts.

(c) Search engines prioritise pages that are frequently updated, which makes sense as old content could imply a company has gone out of business or that information is out of date.

4.1.4 Conclusion

Almost every service you will have used will have been managed by one or more databases; whether it is banking, shopping, social media, travel, logging on to your OU module website, making a phone call, watching an online movie or a thousand other things. It is hard to imagine a world without databases.

Now I've introduced some of the uses for databases, it is time to find out exactly what they are.

4.2 Flat databases

Let's start by looking at the simplest of all databases: the **flat database**, which is sometimes also called a **table**. College_Table (below) is an example of a flat database containing information about some fictional students and the equally fictional evening classes they are studying. In the language of databases, a column in this table can be described as a **field**, and each row is a **record**. This table is quite short, but it could easily be extended downwards to cover many more students.

College_Table

Name	*Date_of_Birth*	*Address*	*Postcode*	*Course_1*	*Course_1_Weeks*	*Course_2*	*Course_2_Weeks*	*Course_3*	*Course_3_Weeks*
Lodhi, Mona	25/05/1979	22 The Grove, Newport	AB12 3CD	Yoga	10	Holiday Spanish	15		
Jones, Bobby	30/11/1990	40 Eldon Court, Hampton	AB6 4PQ	IT for all	5	Oil painting	15		
Cherry, Colin	14/12/1968	59 Acacia Avenue, Brompton	AB2 12ZY	Digital photography	10	Sewing	15		
Cherry, Colin	03/01/1999	8 Mount Pleasant, Greenhill	AB7 4PP	Creative writing	20				
Edwards, Delia	17/08/1987	40 Eldon Court, Hampton	AB6 4PQ	Digital photography	10	Oil painting	15	Ballroom dancing	30
Roberts, Albert	08/09/1952	2 High Street, Stratford	AB12 7UB	Drawing	15				
Singh, Sara	21/10/1977	7 Marina View, Sutton	AB14 8WQ	Woodwork	15				
Chang, Patrick	17/08/1987	21 Green Lane, Newport	AB12 9TU	Everyday maths	20				
Evans, Mary	08/09/1937	13 The Limes, Leighton	AB6 7LR	Oil painting	15				

Activity 4.3 (self-assessment)

What are the names of the fields in College_Table?

College_Table can store up to three courses for each student using the fields *Course_1*, *Course_2* (where students study a second course) and *Course_3* (for those taking three courses). Course lengths are stored in the fields *Course_1_Weeks*, *Course_2_Weeks* and *Course_3_Weeks*.

Data types

It is all too easy to make errors when entering data into a database, so most databases restrict the **type** of data that can be stored in an individual field. Here are some common types used in databases:

- Text: Any combination of text, numbers and symbols.
- Numbers: Numerical values. Some databases further divide numbers into integers (whole numbers only), floating-point (decimal) numbers and scientific (exponential) notation. It may also be possible to set valid ranges for numbers.
- Date and times: Calendar dates and clock times. Different formats will be available, depending on where the database is to be used and its purpose. The database will automatically check that dates and times are valid.
- Currency: Monetary values in a range of currencies.
- Logical: True or False values only.

The database developer is responsible for choosing appropriate types for each field. For example, a field storing a person's age would be set to use the Number type. It might be further restricted to only accept positive whole numbers and perhaps have a maximum age. If a user then attempts to enter text into the age field, or give someone an age of −1 or 1000, the database would reject the text and perhaps prompt them with an error message telling them what they had done wrong.

Advanced databases support a wider range of data types, including the storage of pictures, video and audio content as well as formatted documents such as ebooks, links to data on the world wide web or other databases, and so on.

Activity 4.4 (self-assessment)

Here is the record for Mona Lodhi. Specify a suitable data type for each field.

Name	*Date_of_Birth*	*Address*	*Postcode*	*Course_1*	*Course_1_Weeks*	*Course_2*	*Course_2_Weeks*	*Course_3*	*Course_3_Weeks*
Lodhi, Mona	25/05/1979	22 The Grove, Newport	AB12 3CD	Yoga	10	Holiday Spanish	15		

As well as using data types to avoid storing incorrect values, the database designer can also program the database with rules to check every value as they are entered. One rule might ensure the student's name contains both a forename and second name; another might check an address and postcode against the Royal Mail's postcode database to confirm they match one another, and so on.

4.2.1 Interacting with databases

Once data has been stored in the database, it is possible to put it to work. The most common task for databases is answering users' **queries**. An English-language query for the student database might be:

> How many students have postcodes beginning 'AB6'?

That question can easily be answered by a human so long as the database isn't much longer than College_Table, but you can easily imagine the tedium and scope for error if the college had tens of thousands of students.

Query languages

Databases manage queries using specialised **query languages** which combine a very limited subset of English phrases with logical commands that allow items to be included or excluded from the query. The most common query language is called the **Structured Query Language (SQL)** – sometimes pronounced 'sequel'. First released by IBM in 1974, SQL is now an international standard supported, at least in part, by most database applications.

SQL's popularity is in large part due to its relative simplicity; a relatively small number of English-like commands allow users to create, access, modify and delete data without requiring any understanding of computer programming. This simplicity was deliberate; SQL was designed to allow business managers to interact with databases at a time when almost all computer users were highly trained specialists.

Here is how the English-language query above would be written using SQL:

```
SELECT * FROM College_Table WHERE Postcode LIKE
'AB6%';
```

This might look intimidating if you've never programmed a computer or used a computer's command line to enter instructions, but the query itself is straightforward and is explained in Table 4.1.

Table 4.1 Explaining commands

Command	Description
`SELECT`	an SQL command to select some data from a database
`*`	means 'every record'
`FROM College_Table`	tells the query to look in College_Table
`WHERE Postcode`	another SQL instruction to look in the *Postcode* field of College_Table
`LIKE 'AB6%'`	to find items beginning '*AB6*'.

The query would look through the entirety of College_Table but only return information for Bobby Jones, Delia Edwards and Mary Evans since they are the only students in College_Table whose postcodes begin with '*AB6*'.

Databases and query languages underpin most websites. Rather than storing web pages for every product in an online store or story in an online newspaper, a typical website stores components of the page – the text, pictures, prices, reviews, and so on – in one or more databases.

Following a request for a web page, the site's server constructs the necessary query to retrieve relevant information from the database. The data is then formatted into an HTML web page and sent to the user. Crucially, users don't need to understand any query language, or even know that query languages exist; instead, queries are automatically created and edited as users follow links, click buttons, select menu items and submit searches.

Box 4.2 Storing results of queries

Many websites store the results of popular queries (such as the front page of a newspaper or their most popular product) as formatted HTML pages and deliver the stored copy to users, rather than using a new query to generate a fresh copy of the page each time it is requested.

This process is called **caching** and can greatly improve the ability of a site to serve many simultaneous users because the site does not need to keep accessing the underlying database. Cached pages are periodically replaced using queries to ensure they are always accurate and up-to-date.

Not only do databases allow ecommerce sites to add new items or news sites to publish new stories at any time, but they also greatly reduce the risk of error when these changes are made. Without a database, if a piece of data, such as a telephone number, is changed, every page using that data must be identified and corrected. Clearly this becomes increasingly challenging as the site gets ever larger. If instead, the telephone number is part of a database record, then only that entry in the database needs to be changed. As soon as the correction is made, every page referencing the record automatically reflects the change.

Activity 4.5 (exploratory)

This activity gives you an opportunity to interact with a database using a query language. Don't worry – we'll start by creating queries using on-screen buttons and other widgets.

On the module website you will find this database along with specific instructions for completing this activity.

4.2.2 The limitations of flat databases

Our simple database appears to do a good job: it is easy to find individual students and the courses they study. Although we haven't yet implemented it as a computerised database, it would be relatively simple to do so, and even a modest PC could handle a flat database containing tens of thousands of student records. If everything remained unchanged, this database would be perfectly adequate for our purposes. However, we'd quickly encounter problems if we added new data or needed to change values of previously stored data.

Constructing a database requires careful thought about the way information is organised and used – something I didn't do when I created the original flat college database.

Adding fields

Delia Edwards finds she has more time on her hands and wants to study woodwork as well as the three courses she is already taking. The current table can only accommodate students taking three courses, so we would have to add another field to accommodate Delia. But is four sufficient? Perhaps we should add a fifth course, just in case Delia gets bored – but how many more fields should there be?

We could add an extravagant number of fields, such as 20 courses, and we could be reasonably confident, if not entirely certain, that no one would ever exceed that number of courses in a year. But such a table is wasteful since much of it would be empty. This is problematic because:

- empty fields occupy memory since the computer must store the structure of the database, whether its fields are used or not
- queries run more slowly on large tables than smaller tables, even when largely empty, since there is more data to search through.

Of course, we could just use a faster computer to overcome these deficiencies – but there is a second problem which cannot be solved by more processing power.

Editing records

The second concern is that we are repeating data in our database. This has a minor drawback in that it uses some extra memory, but there is a more serious issue whenever information is repeated.

Activity 4.6 (self-assessment)

Can you think of a reason why repeating data in a database might be problematic? The data in College_Table might be helpful in formulating your answer.

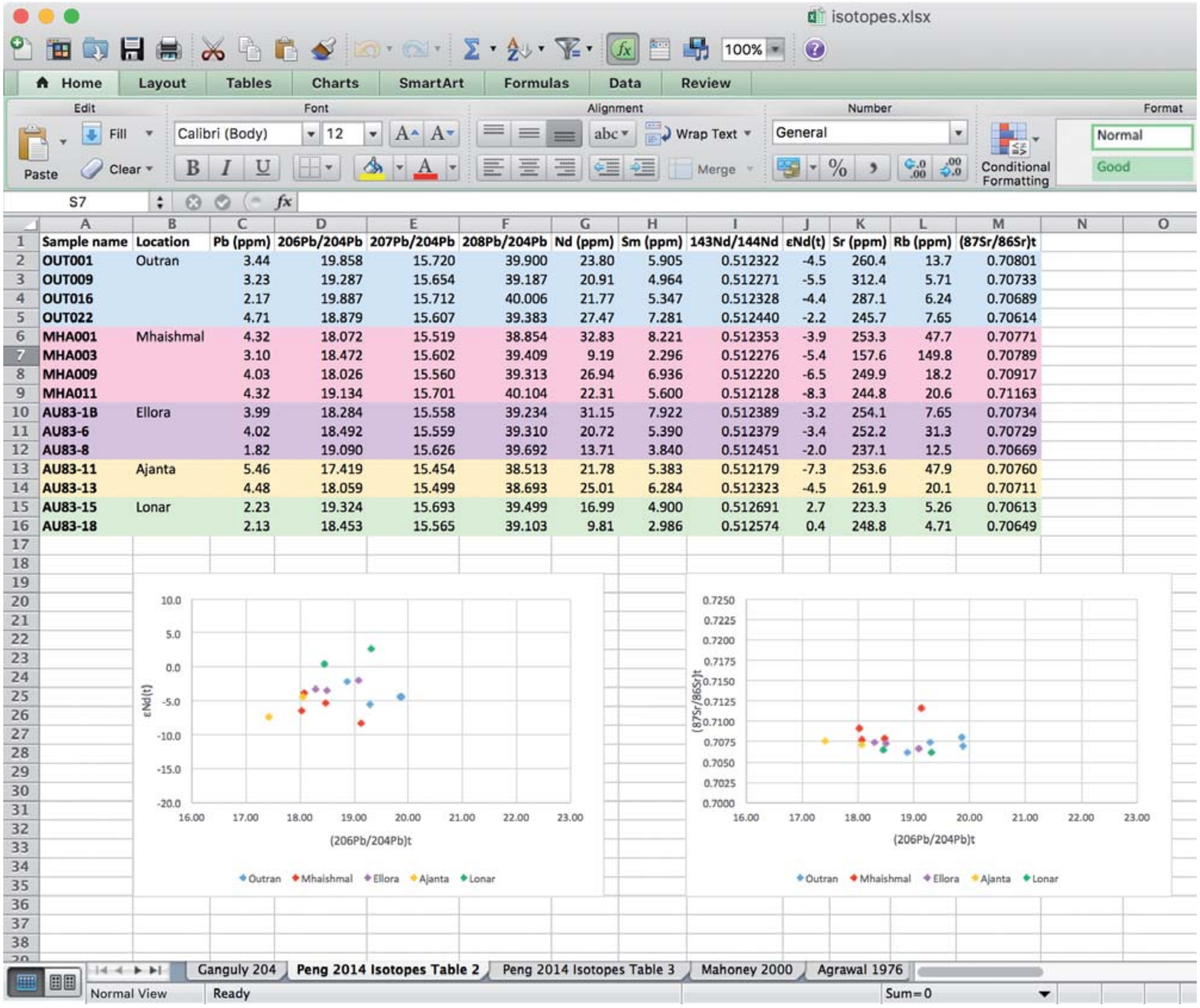
Changing data or correcting errors requires finding and editing every repetition in the table. For instance, following enthusiastic feedback from a previous group, *Digital photography* will be extended from 10 to 15 sessions. Incorporating this amendment in College_Table requires updating several student records. We can't simply rely on search and replace to alter every occurrence of 10 to 15, since *Yoga* also runs for 10 weeks and its length does not need to change. Instead, we need to check every record in the database – a time-consuming and error-prone process.

Ideally, information about the number of sessions would be recorded in just one place, and editing *Digital photography* would involve changing a single record. This is the principle behind the so-called **relational database**, which we will explore in the next section.

4.2.3 Spreadsheets and databases

Databases are often confused with spreadsheet applications (such as Microsoft's Excel program). This is understandable; both are concerned with storing and processing huge amounts of data, and the flat college database shown above might well remind you of a spreadsheet.

Databases *can* be used like spreadsheets to calculate totals, generate statistics and process data using equations to generate new values. Similarly, a spreadsheet *can* be used to store more than just numeric information. Figure 4.4 shows an example of a Microsoft Excel spreadsheet which stores geological data, including the names of samples, their locations and a range of geochemical values.



Sample name	Location	Pb (ppm)	206Pb/204Pb	207Pb/204Pb	208Pb/204Pb	Nd (ppm)	Sm (ppm)	143Nd/144Nd	εNd(t)	Sr (ppm)	Rb (ppm)	(87Sr/86Sr)t
OUT001	Outran	3.44	19.858	15.720	39.900	23.80	5.905	0.512322	-4.5	260.4	13.7	0.70801
OUT009		3.23	19.287	15.654	39.187	20.91	4.964	0.512271	-5.5	312.4	5.71	0.70733
OUT016		2.17	19.887	15.712	40.006	21.77	5.347	0.512328	-4.4	287.1	6.24	0.70689
OUT022		4.71	18.879	15.607	39.383	27.47	7.281	0.512440	-2.2	245.7	7.65	0.70614
MHA001	Mhaishmal	4.32	18.072	15.519	38.854	32.83	8.221	0.512353	-3.9	253.3	47.7	0.70771
MHA003		3.10	18.472	15.602	39.409	9.19	2.296	0.512276	-5.4	157.6	149.8	0.70789
MHA009		4.03	18.026	15.560	39.313	26.94	6.936	0.512220	-6.5	249.9	18.2	0.70917
MHA011		4.32	19.134	15.701	40.104	22.31	5.600	0.512128	-8.3	244.8	20.6	0.71163
AU83-1B	Ellora	3.99	18.284	15.558	39.234	31.15	7.922	0.512389	-3.2	254.1	7.65	0.70734
AU83-6		4.02	18.492	15.559	39.310	20.72	5.390	0.512379	-3.4	252.2	31.3	0.70729
AU83-8		1.82	19.090	15.626	39.692	13.71	3.840	0.512451	-2.0	237.1	12.5	0.70669
AU83-11	Ajanta	5.46	17.419	15.454	38.513	21.78	5.383	0.512179	-7.3	253.6	47.9	0.70760
AU83-13		4.48	18.059	15.499	38.693	25.01	6.284	0.512323	-4.5	261.9	20.1	0.70711
AU83-15	Lonar	2.23	19.324	15.693	39.499	16.99	4.900	0.512691	2.7	223.3	5.26	0.70613
AU83-18		2.13	18.453	15.565	39.103	9.81	2.986	0.512574	0.4	248.8	4.71	0.70649

Figure 4.4 A spreadsheet of geological data which are being charted underneath

The difference between the two types of software lies in their intended purposes. Whilst databases were *always* intended to store different types of data, spreadsheets were originally software recreations of business ledgers used to keep track of company accounts. As such, spreadsheets recreated tasks commonly performed by accountants – initially doing little more than summarising columns or rows of numbers. It was only later, during the late 1970s, that spreadsheets included formulae to calculate profits, losses and taxes, as well as statistical analyses and charting.

Despite all these advances, the fundamental purpose of spreadsheet applications remains their ability to process large volumes of numeric data; therefore, spreadsheets have no need for queries. So, whilst it is possible to search for individual values within a spreadsheet, it is not possible to replicate the complex selections and modifications to data performed by database queries.

4.2.4 Conclusion

This section introduced you to some of the key concepts in databases and allowed you to see how organisations can store large amounts of data in tables by assigning attributes to this data. You also learned how data is added to, modified and retrieved from databases by query languages. Finally, you saw that whilst flat databases are a major advance on storing documents on paper, they have intrinsic problems affecting their usability and reliability.

4.3 Building large databases

Relational databases overcome the problems outlined previously by dividing data between two or more tables. This separation is known as **normalisation**, during which the flat database is reorganised to identify related data as well as unwanted or duplicated data.

Let's explore normalisation by converting the flat college database into a relational database.

4.3.1 Entities and attributes

I first need to introduce the concept of an **entity**. An entity is any item about which we want to store information and is the fundamental building block for relational databases. Entities can be described as *tangible*, such as a person, a car or an album; or *intangible*, such as an invitation to an event.

Returning to College_Table, from Section 4.2, the most obvious entity is a *Student* – after all, what is a college without students? Students, because they are physical objects, are clearly examples of tangible entities. An individual student, such as Colin Cherry, is an **instance** of the *Student* entity; Sara Singh is another instance of *Student*, as is Delia Edwards, and so on.

Entities are described using **attributes** – so the attributes of a student can be their name, date of birth, address, and so on. Whilst Colin Cherry and Sara Singh are both instances of the *Student* entity and have the same attributes, their attributes have different **values**.

Activity 4.7 (self-assessment)

(a) What are the attributes of a student entity in College_Table?

(b) What are the values for attributes belonging to Albert Roberts?

You could argue that *Course_1*, *Course_2* and *Course_3* are also attributes of *Student* because students study courses. However, an individual course is a collection of information possessing a name and duration in weeks. Courses therefore have attributes, so we should consider *Course* to be an intangible entity.

Activity 4.8 (self-assessment)

Using the information in College_Table, should we also treat *Postcode* as an entity?

The next stage of creating the relational database is to divide the data into separate tables with each table representing a single entity. Since there are two entities in College_Table, we need two tables.

Student_Table is the entity table for the *Student* entity. Course_Table is the corresponding entity table for the *Course* entity. To make the tables slightly more realistic, student names have been split into family and given names. The *Course* entity now stores a name, the number of sessions and its cost.

Each table is structured just like a flat database, consisting of horizontal records and vertical fields. A record holds data about a single instance of an entity; a field stores data about one attribute of those entities. Here are the two new tables.

Student_Table

Family_Name	*Given_Name*	*Date_of_Birth*	*Address*	*Postcode*
Lodhi	Mona	25/05/1979	22 The Grove, Newport	AB12 3CD
Jones	Bobby	30/11/1990	40 Eldon Court, Hampton	AB6 4PQ
Cherry	Colin	14/12/1968	59 Acacia Avenue, Brompton	AB2 12ZY
Cherry	Colin	03/01/1999	8 Mount Pleasant, Greenhill	AB7 4PP
Edwards	Delia	17/08/1987	40 Eldon Court, Hampton	AB6 4PQ
Roberts	Albert	08/09/1952	2 High Street, Stratford	AB12 7UB
Singh	Sara	21/10/1977	7 Marina View, Sutton	AB14 8WQ
Chang	Patrick	17/08/1987	21 Green Lane, Newport	AB12 9TU
Evans	Mary	08/09/1937	13 The Limes, Leighton	AB6 7LR

Course_Table

Course	*Sessions*	*Fee*
Yoga	10	20
Digital photography	10	20
Everyday maths	20	40
Oil painting	15	30
Creative writing	20	40
Holiday Spanish	15	30
Woodwork	15	30
Drawing	15	30
Sewing	15	30
Ballroom dancing	30	60

4.3.2 Keys

An important piece of information was lost during the transition from the original flat database to a relational database. The tables no longer show which students are registered on which courses – the entire point of the original database! This can be fixed by linking each student in Student_Table to one or more courses in Course_Table. Before the two can be linked, every entity in each table must be identified using a unique **key** field. Sometimes a suitable key field will already exist in the table, but as you can see from Student_Table, this is not always the case.

Activity 4.9 (self-assessment)

Can you identify why we cannot use names, dates of birth or addresses in Student_Table as unique keys?

In this case, it is necessary to create a new key called *Student_Number* – a sequential numbering of student records. Numbering records is by far the simplest way of creating a key when no suitable attribute will do. A real-world example is your Open University personal identifier (e.g. A123456), which acts as a key value in OU student databases.

The new *Student* entity table is shown below, with names and addresses ordered by ascending student number. The key field itself – *Student_Number* – is shown in bold.

Student_Table

Student_Number	*Family_Name*	*Given_Name*	*Date_of_Birth*	*Address*	*Postcode*
000001	Lodhi	Mona	25/05/1979	22 The Grove, Newport	AB12 3CD
000002	Jones	Bobby	30/11/1990	40 Eldon Court, Hampton	AB6 4PQ
000003	Cherry	Colin	14/12/1968	59 Acacia Avenue, Brompton	AB2 12ZY
000004	Cherry	Colin	03/01/1999	8 Mount Pleasant, Greenhill	AB7 4PP
000005	Edwards	Delia	17/08/1987	40 Eldon Court, Hampton	AB6 4PQ
000006	Roberts	Albert	08/09/1952	2 High Street, Stratford	AB12 7UB
000007	Singh	Sara	21/10/1977	7 Marina View, Sutton	AB14 8WQ
000008	Chang	Patrick	17/08/1987	21 Green Lane, Newport	AB12 9TU
000009	Evans	Mary	08/09/1937	13 The Limes, Leighton	AB6 7LR

For Course_Table, the names of the courses could be used as keys. However, just in case more than one course has the same name but is taught at different levels, each course can be allocated a unique *Course_Code* to act as a key. (Again, OU module codes such as TM111 are key values in the university databases.) Our revised Course_Table is shown below, organised by course code (in bold).

Course_Table

Course_Code	*Course*	*Sessions*	*Fee*
ACT01	Yoga	10	20
ACT02	Ballroom dancing	30	60
CRT01	Digital photography	10	20
CRT02	Oil painting	15	30
CRT03	Creative writing	20	40
CRT04	Woodwork	15	30
CRT05	Drawing	15	30
CRT06	Sewing	15	30
LNG01	Holiday Spanish	15	30
MAT01	Everyday maths	20	40
TEC01	IT for all	5	10

Activity 4.10 (self-assessment)

Course_Table uses the course code as a key. How valid is the course code as a key under the following circumstances?

(a) Courses are presented only once, and never repeated.

(b) Courses are run only once per year, and are repeated the following year.

(c) Some courses are run twice or more per year.

4.3.3 Joining things up

The next stage of the design is to re-establish **relationships** between the *Student* and *Course* entities using the new key values. It is these relationships that give their name to relational databases.

Relationships are stored in a so-called **joining table** listing every relationship between every student and their course (or courses). The following joining table, Student_Allocation_Table, shows links between students (on the left) and courses (on the right).

Student_Allocation_Table

Student_Number	*Course_Code*
000001	ACT01
000001	LNG01
000002	TEC01
000002	CRT02
000003	CRT01
000003	CRT06
000004	CRT03
000005	CRT01
000005	CRT02
000005	ACT02
000006	CRT05
000007	CRT04
000008	MAT01
000009	CRT02

Each record in the joining table represents a single relationship between a student and a course. You will notice that some student keys appear more than once in the left-hand column, and some courses appear more than once in the right-hand column.

- A student key appearing more than once shows an individual student studies *many* courses.
- A course code that is listed multiple times shows the course is being taken by *many* students.

Database designers call this a **many-to-many relationship**. The entity tables and joining table taken together constitute the relational database. A real-life relational database is likely to have many entity tables and many more joining tables.

Activity 4.11 (self-assessment)

Using Student_Allocation_Table:

(a) Identify a many-to-many relationship where a student is studying more than one course. Use the key values in the joining table to find the name of the student and their courses from the other tables.

(b) Identify another many-to-many relationship where a course is being taken by more than one student. Use the key values to name the course and its students.

Note, the joining table is not immediately useful to humans; it is impossible to say from Student_Allocation_Table alone what the names of the students are, their addresses or any information about courses. It is important to remember that the joining table is for the convenience of the computer – not for humans.

We can see how all these tables interact by imagining a query to find out which courses are being taken by Delia Edwards.

1 The user would enter 'Delia Edwards' into an application that can perform queries on the database. This might be a college web page linked to the database or the database management application itself.
2 This generates a query to Student_Table obtaining Delia's *Student_Number* (000005).
3 The database then makes a second query. Delia's *Student_Number* is passed to the Student_Allocation_Table joining table which returns every record where the *Student_Number* field contains 000005. Delia is studying courses with *Course_Code*s of CRT01, CRT02 and ACT02.
4 A succession of further queries is made to Course_Table using each *Course_Code* in turn to obtain the human-readable titles of the courses so that they can be displayed to the user.

This is illustrated in Figure 4.5.

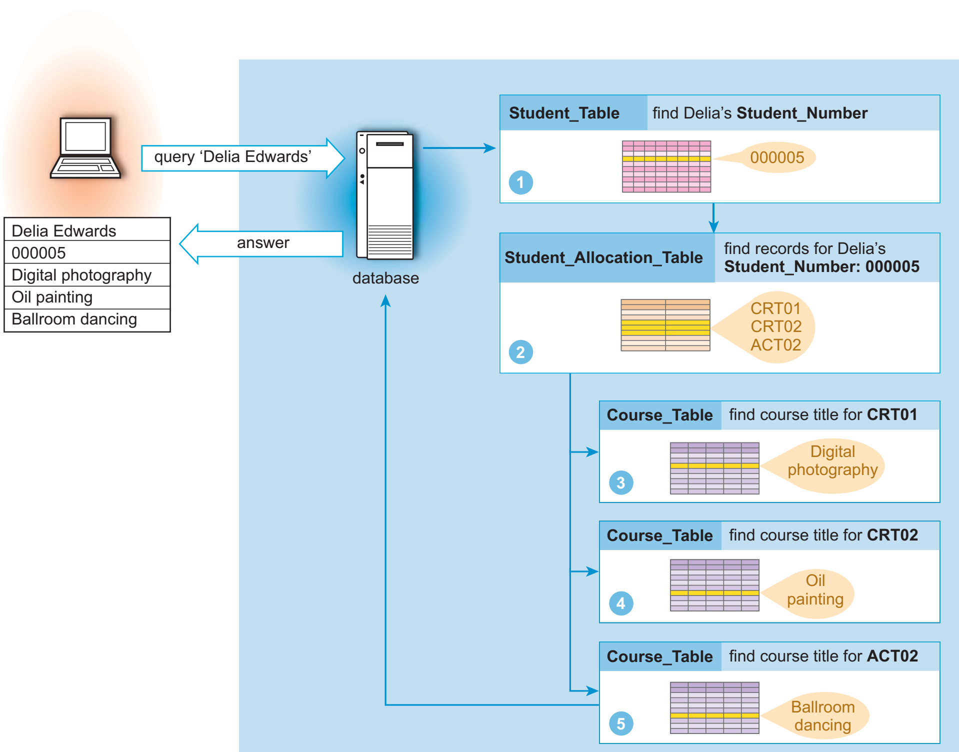


Figure 4.5 Queries and results

Apart from the initial query, every subsequent query happens 'behind the scenes' in a tiny fraction of a second. The complexity of the database is entirely hidden from view, which is perhaps how it should be – although it does mean the database designers don't always get the recognition they deserve.

Activity 4.12 (exploratory)

The following table represents a flat database relating to some regular committee meetings at the college. The college's computer systems are being redesigned and the existing database is going to be converted into a relational database.

Meeting_Table

Committee_Name	*Meeting_Time*	*Member_Family_Name*	*Member_Given_Name*	*Extension*
Planning	Weekly	Jones	Jennifer	1239
		Patel	Amala	4728
		Robinson	Sarah	1000
Recreation	Monthly	Jones	Jennifer	1239
		Smith	Teresa	1000
Education	Quarterly	Patel	Manish	4401
		Robinson	Sarah	1000

After studying the table, work through the following points.

(a) This table represents two entities. What are the entities and their associated attributes?

(b) Create two new tables representing these entities. Identify a key field in each, or if there is not a suitable existing field, create a new key field.

(c) Assume the existing flat database must be normalised into separate tables for the *Committee* and *Member* entities. Why is a joining table needed?

(d) Construct the joining table, using the key from *Committee* as one key and the key from *Member* as the other. You can write your answer on paper or complete it in a spreadsheet or text document, but make sure you attempt it *before* revealing the answer.

Comment

(a) The table contains data about two distinct entities: committees and members of staff.

(i) *Committee*: the associated attributes are *Committee_Name* and *Meeting_Time*

(ii) *Member*: the associated attributes are *Member_Family_Name*, *Member_Given_Name* and *Extension*.

(b) My two tables are shown below. (The key field is shown in bold.) The names of committees are unique so they can act as key values for Committee_Table. There are no suitable existing keys for Member_Table – the college employs two Patels, whilst Sarah Robinson and Teresa Smith share an extension number. We could use *Member_Given_Name* as a key since no two members of staff currently have the same first name, but there is every likelihood that sooner or later two members of staff will have the same first name, which would invalidate its use as a key. Instead we have created a new key, *Staff_Number*.

Committee_Table

Committee_Name	*Meeting_Time*
Planning	Weekly
Recreation	Monthly
Education	Quarterly

Member_Table

Staff_Number	*Member_Family_Name*	*Member_Given_Name*	*Extension*
001	Jones	Jennifer	1239
002	Patel	Amala	4728
003	Robinson	Sarah	1000
004	Smith	Teresa	1000
005	Patel	Manish	4401

(c) A joining table is required because the relationship between *Committee* and *Member* is many-to-many: a committee has several members, and each member can be on more than one committee.

(d) The joining table:

Meeting_Allocation_Table

Committee	*Member*
Planning	001
Planning	002
Planning	003
Recreation	001
Recreation	004
Education	005
Education	003

4.3.4 The benefits of normalisation

The process of normalisation has converted the college's single table flat database into a relational database. We have created two new tables each storing data about a distinct entity and linked them using a joining table which stores relationships between individual students and their courses. Normalisation has solved both problems identified with the original design:

1 If a student enrols for several courses, new records can be added to the Student_Allocation_Table joining table. Each record in Student_Allocation_Table therefore represents the enrolment of a student on one course. It is not necessary to have empty fields in which to store data for students taking more than one course, as we did in the original College_Table.
2 If there is an administrative change, e.g. changing a course fee, or a student changing their address, then only a single record needs to be changed in the appropriate table. You will recall repetition of data in the original database meant that changes required hunting out every occurrence of the data and modifying it.

Normalisation allows data to be efficiently organised into tables. Unnecessary repetition has been eliminated and each table now represents a single entity and its attributes. After performing normalisation on the database, the original flat database can be criticised using more appropriate language.

The flat database is inadequate because:

- it contains data about more than one entity
- it attempts to capture the relationship between entities.

Many database applications automate normalising data and creating the necessary joining tables; but for very large databases, the process of designing a suitable database structure is a highly skilled job.

4.3.5 Views

A key difference between databases and spreadsheets is that databases permit different **views** of the data. A view uses a query to combine data from one or more tables to produce a new **virtual table** which can itself be queried by the user. Views are not fixed in the database's structure; instead, they are created as and when they are needed.

Views have many uses including:

- Allowing data from multiple tables to be displayed alongside one another, hiding the complexity of the underlying database.
- Limiting access to critical information. In our college database, a lecturer would want to know the names of the students on their course, but, in the interests of the student's privacy, we should provide lecturers with a view hiding the student's home address from casual inspection.
- Generating data on demand by combining results from several different tables. Our college could create a custom view for students containing an itemised bill for their studies by combining data from their list of courses and another table listing prices.
- Increasing security. Views can be designed to be 'read only', allowing users to see data, but not change it. Students could look at the curriculum

through a view of current courses and see the cost of each course, but they would not be allowed to edit any of the data.

Database views have a couple of drawbacks that are briefly worth considering:

- Views are dependent on the arrangement of the underlying tables. If the database design is altered, then views will also need to be changed. Fortunately, this should be an infrequent occurrence – a good database design shouldn't need to be redesigned unless its purpose is radically changed.
- Accessing data through views tends to be slightly slower than directly accessing the tables, so performance can suffer when views are complex.

Almost any online service that allows you to customise what you see, or offers information that it thinks you will find useful, will be using views onto a common database. Views will be constructed using software that analyses your past behaviour as well as habits of users it considers are like you.

4.3.6 Database management systems

Formally, the software hosting a database and with which users and other applications interact is known as a **database management system (DBMS)**. Confusingly, the DBMS is often just called 'a database'. A relatively small number of companies dominate the market for DBMSs, each with their own designs and functionality, although they share most of the same fundamental principles. DBMSs range from applications designed to provide basic database services to single users, such as Microsoft Access, through to systems such as IMS and Oracle databases which can perform thousands of simultaneous queries across multiple databases distributed around the world.

Common DBMS types described on websites or in job advertisements include Microsoft Access, Oracle, IBM DB2, Microsoft SQL Server and MySQL. Generally, databases designed in one DBMS are not easily ported to another DBMS, but they can be made to interoperate, either by making one DBMS compatible with another or by supporting industrial standards such as the SQL language mentioned earlier.

Box 4.3 Standards

Standards are tightly-defined descriptions of how a product should be constructed or operate. SQL is an example of an International Organization for Standardization (ISO) standard. SQL is defined in a series of standards documents known as ISO/IEC 9075, so any company wishing to offer support for SQL in their products should follow *all* the requirements in ISO/IEC 9075. In actuality, most DBMS providers do not exactly conform to this standard, so different implementations of SQL may be incompatible with one another.

The difficulty of transitioning between DBMSs means that databases have very long lives, often lasting many decades. These old but still highly-useful databases are examples of what are known as **legacy systems**. There is a clear implication in this phrase that legacy databases are not as good as modern databases. In fact, there is no reason to stop using a legacy system as long as it can meet the needs of its users and can be maintained. The risk of legacy systems lies largely in that they often run on obsolete computer systems, or that the company that wrote the DBMS has either stopped all development and support of the software or has ceased trading entirely.

4.3.7 Conclusion

This section has shown how the relational database allows us to build databases of much greater scale than before, whilst managing data in such a manner that we can eliminate unnecessary duplication and more clearly understand our data.

Whilst relational databases are more complicated than flat databases, their benefits far outweigh the additional time and effort required in their creation. Any extra time spent in designing a relational database is more than offset by the reduction in time needed to maintain the database. Unsurprisingly, almost all real-world databases are implemented using relational database technologies.

However, our ability to create, store and process data now outstrips the capabilities of relational databases. I am going to finish this part by exploring the newest and perhaps most exciting application for data: big data.

Big data

4.4

Databases have been a fundamental tool for organising data for more than fifty years and they will continue to be useful into the foreseeable future. However, databases were invented when data was relatively rare and expensive to acquire, and when data could be neatly categorised into entities.

That is no longer the case; in 2013, the ACI Information Group estimated that humanity had created 5 exabytes of data between the dawn of civilisation and 2003. (One exabyte is 1×10^{18} bytes, which is one billion gigabytes. It is abbreviated to EB.) That 5 EB included every book, song, television programme, radio broadcast, movie, inscription, letter, note – everything by everyone who had ever lived.

By 2015, humanity was generating 5 EB of data every 2 days! And the rate at which data is acquired continues to accelerate.

This incredible growth in data results from the widespread digitisation of society. Traditional media have been replaced by digital storage and transmission, whilst new industries including social media and the 'Internet of Things' are built around creating and sharing vast amounts of new data. So not only are we creating more data, but that data no longer falls into the neat categories suited to conventional databases. A new type of data-handling technology is needed – one designed to extract useful information from huge pools of ill-defined data.

It goes by the generic name of '**big data**'.

Activity 4.13 (exploratory)

You may not have heard of big data before now, but it is already an important part of the modern world. On the module website, there is a short video from BBC *Horizon* showing how big data is being used by police in Los Angeles to identify where resources should be targeted to prevent crime. When watching this video, look for:

- the ways in which data about historic crime data is analysed
- how the police respond to the analysis.

Comment

Analysis of historic crimes identifies regions of the city which are most prone to crime as well as times and dates when crime increases. Police can be deployed to these areas before damage and injury occurs. None of this would be possible without the ability to process enormous amounts of data extremely rapidly.

4.4.1 Sampling

Traditionally, data processing has been so complicated and time-consuming that it is only possible to examine a subset of the available data, known as the **sample**. To give a simple example, if transport researchers wanted to learn more about traffic congestion in a city, they *could* record the movement of every car passing through the city. Obviously, this would be extremely complicated and expensive, so instead researchers monitor *some* journeys throughout the day. If their sample of car journeys is representative of all journeys, the researchers can determine which routes and times are most prone to congestion and perhaps suggest methods of mitigating traffic problems.

Sampling has been used for a wide range of scientific, political and economic purposes for more than two centuries. Perhaps the most familiar samples are political opinion polls. In the UK, a typical poll will use a sample of approximately 1000 people selected from an electorate of 45 million people. The sample will have been carefully chosen to reflect the wider electorate in terms of age, employment status, income, ethnicity, if they are rural or urban dwellers, and so on. If the sample is representative –and if the members of the sample do not lie to the pollsters – it *should* be able to predict voting intentions accurately.

However, experience has shown that creating a representative sample is incredibly hard; the sample can contain hidden biases or omissions that will lead to mistaken conclusions. A well-studied example of this failure was the predictions of the outcome of the 2015 UK general election. Prior to the election, the clear majority of polls predicted a hung parliament in which no single party would have the majority needed to form a government. Furthermore, most polls gave the opposition Labour Party a slight lead over the governing Conservative–Liberal Democrat coalition, suggesting that Labour would be the largest party in the House of Commons. Figure 4.6 illustrates the results from published opinion polls for the 5 years leading up to the 2015 UK general election. In the event, the Conservatives achieved a slight majority and formed a majority government without entering coalition.

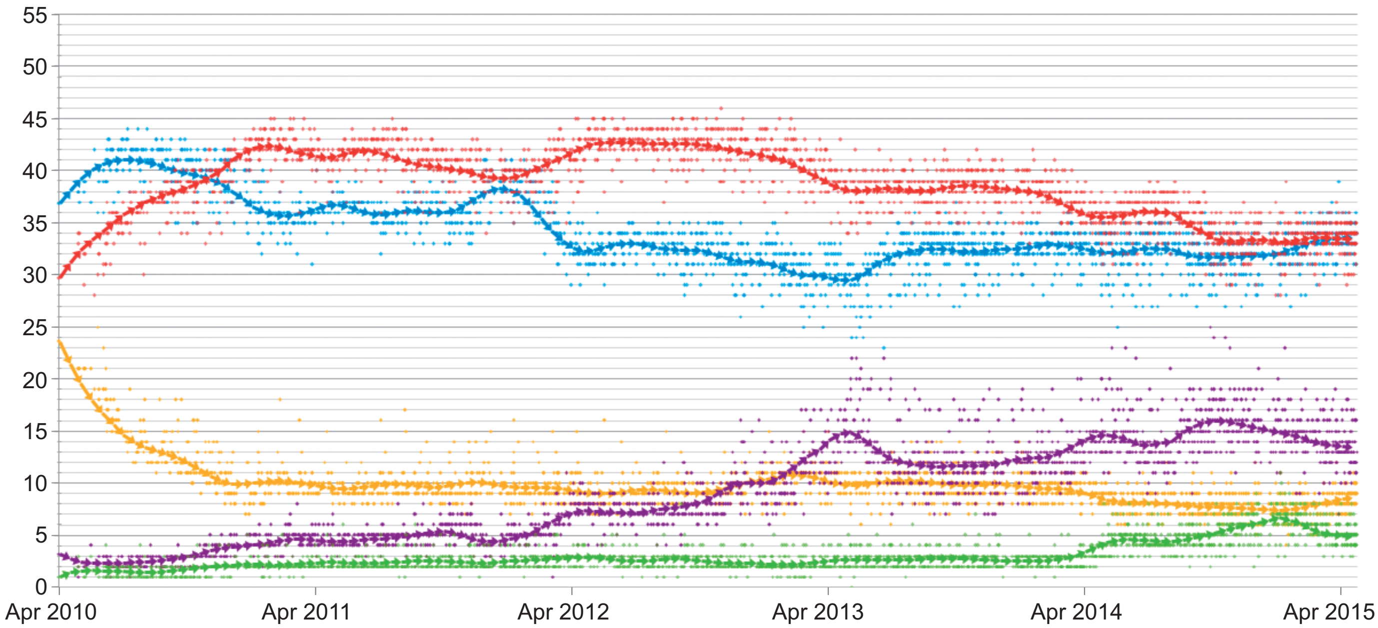


Figure 4.6 Each dot represents a value from a single poll, with the lines showing the trend for each party, starting with the Conservative Party (blue) at the highest point, followed by the Labour Party (red), Liberal Democrats (yellow), the United Kingdom Independence Party (purple) and the Green Party (green). Smaller parties and regional parties are not shown.

The failure of hundreds of opinion polls to predict this result led polling companies to conduct an urgent review into their methods. The error could not be explained by a 'late swing' to the governing party, or that Conservative voters were more likely to vote than Labour supporters. Instead, the review found that pollsters inadvertently, but systematically, overrepresented Labour voters in their samples at the expense of Conservative voters, with the result that Labour appeared more popular than was in fact the case in the entire electorate.

Sampling has a further drawback in that both the sample and the data obtained from the sample are defined at the start.

- A political opinion poll might be conducted using landline telephones, so that pollsters can be sure of the geographical spread of respondents; but in doing so it will exclude that part of the population, disproportionately young, who don't have landlines and who might have different voting intentions. Conversely, an internet-only poll has the opposite bias, favouring younger voters.
- Similarly, pollsters cannot ask questions that are not in the poll, even if they realise that a key question is missing. Their only choice would be to commission a second poll. This is not only expensive, but there is a chance that the two samples are sufficiently different that the results are not comparable.

Sampling is not limited to politics; it is a key method in scientific investigation, where the time and expense of processing data has traditionally forced researchers to examine only a tiny subset of the possible data.

Box 4.4 Genetic testing and sample size

23andMe sells a genetic testing kit costing £125. The customer uses a swab to collect cells from the inside of their cheek and sends it to the company for analysis. A few weeks later, they receive a personalised report informing them if they have one of more than 40 inherited medical conditions, sensitivities to certain medications, and a genetic component to some of their physical characteristics. The test is relatively cheap since 23andMe only samples a relatively small number of genes known to be associated with certain traits and conditions. Diseases associated with genes outside this sample, or those whose genetic origin is uncertain, cannot be detected.

23andMe can provide affordable testing by restricting its sample size, but with the drawback that if a new test is developed for another medical condition, customers must send new specimens for analysis. By contrast, a complete DNA sequence for a user avoids the problem of limited sampling; once sequenced, a complete digital record of their DNA can be stored and analysed whenever new genetic links to disease are discovered. The trade-off is that analysing complete sequences is computationally intensive and more expensive - although that price has dropped from US$2.7 billion per person in 2003 to approximately US $1,000 in 2016.

4.4.2 Beyond sampling

Sampling was a product of a time when obtaining and processing data was expensive, both in terms of time and money. Modern computer technology means that the cost of acquiring, storing and processing data has fallen dramatically. Rather than sampling a small subset of the data, it is now possible and affordable to acquire data on the off-chance it will be useful; and then either examine extremely large samples, or indeed process all the data. Big data analyses huge pools of data to find new connections that would have been missed by randomly sampling a tiny fraction of all data.

An existing application for big data is protecting credit card users from fraud. Fraud accounts for a tiny fraction of all transactions, so it is highly unlikely a sample of transactions taken from the many millions of transactions made every day would include a single case of fraud. However, by examining all transactions for unusual patterns of activity, potentially fraudulent transactions can be identified for further investigation.

Box 4.5 Fraud detection

The financial company Xoom specialises in remittances: money sent by immigrant workers to their home countries. This is a business which has traditionally been vulnerable to fraud and money laundering. Xoom's automated systems scrutinise every one of the company's transactions. In 2011, they identified a string of payments originating in New Jersey made on Discover credit cards. None of the transactions were themselves unusual, but the pattern created by the times of the transactions, the sums transferred and their recipients were sufficient to arouse suspicion. Xoom's fraud detection team were alerted and found the transactions involved criminal activity; the payments were blocked, no customers lost money and the criminals were identified.

4.4.3 The three Vs

Big data is sometimes defined using three criteria, called **the three Vs**, originally laid out in 2001 by the data analyst Doug Laney. I have already introduced the first two.

1. **Volume**: very large amounts of data need to be processed – in the case of Xoom, it is all transactions passing through their systems.
2. **Velocity**: there is a relentless demand for data to be acquired and processed at an ever-increasing pace. For Xoom, velocity is crucial – if they can identify fraud quickly, customers are much less likely to lose money and fraudsters have less time to cover their tracks, making identification easier.

Activity 4.14 (self-assessment)

(a) Google Maps (link provided on the module website) lets you view traffic flows with roads colour-coded from green (smoothly flowing traffic) to red (heavy congestion). These colours are generated by tracking the movements of tens of thousands of Android phones whose users have enabled location services such as GPS or Wi-Fi tracking. Participating phones anonymously report their positions every few seconds, from which Google can determine their location, direction and speed. Which of the two Vs do you think apply to Google Maps traffic flows?

(b) Looking at Figure 4.7, do you think YouTube represents a big data project in terms of the two Vs you have already met?

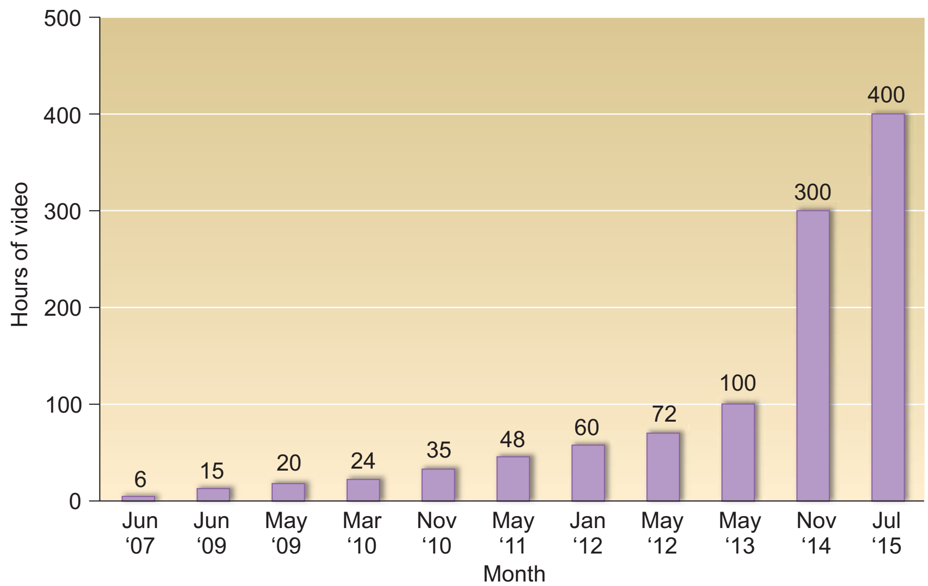


Figure 4.7 Chart showing the number of hours of video uploaded to YouTube every minute from June 2007 until July 2015

The third V is:

> **Variety**: the data being acquired can take any form – not just numeric or textual data, but also video, audio, social media connections and so on. Variety requires us to consider another difference between big data and the data we have previously stored in databases.

4.4.4 Structured and unstructured data

Databases hold information in the form of **structured data**, in which individual pieces of data have clearly defined attributes whose types are known in advance. So, an address stored in a database table could be structured into fields like this:

Address_Table

House_Number_or_Name	*Street_Name*	*Town_or_City*	*County*	*Postcode*	*Country*
Danemead Cottage	Old Pasture Lane	St. Mary Mead	Downshire	MM21 6BY	UK

If I know the structure of the database, I can create a query to extract an attribute (e.g. the county name) from the table. If I also know the data type, I will know what sort of processing I can perform on that data.

By contrast, big data is largely concerned with **unstructured data** – data that has either not been structured in advance, or is unsuited to structuring. Example applications for unstructured data might be identifying a house or person from a photograph or finding stories in a digitised newspaper archive. Unstructured data is produced by both computers and humans:

- **machine-acquired data**, such as satellite and aerial photography images used for online mapping services, scientific data and sensor data
- **human-generated content**, including business documents, emails and social media content.

Note: unstructured does not refer to how data is stored; rather, it refers to the contents of the file itself.

In the last few years, the demand for unstructured data has begun to outstrip that for structured data. A 2014 study by International Data Corporation, a consultancy to the data processing industry, estimated that unstructured data applications used 69 EB of the 80 EB of the data storage shipped that year, and the proportion of storage and processing devoted to unstructured data continues to increase.

Activity 4.15 (exploratory)

Horizon was amongst the first television programmes to investigate the revolutionary potential of big data to transform individual lives. An excerpt from this programme is provided on the module website. It discusses the following three uses for big data that are relevant to our everyday lives.

1. Targeted advertising in online services such as social media, as well as recommended products on shopping sites. The ability to know which products appeal to you is performed using big-data-processing algorithms that began life managing space missions.
2. Analysing how patients use their mobile phones to identify changes in their behaviour which may be linked to changes in their health.
3. The potential to use big data to find links between genetic abnormalities and a range of health conditions. This is very much the cutting edge of modern medicine, but as you saw in the 23andMe example above, some genetic tests are becoming cheap enough that they can be made available to almost anyone.

Watch this video considering the relevance of the three Vs and unstructured data to the problems.

Comment

The topic of personal data is one that inevitably causes controversy because we all have our own feelings about what is, and what is not acceptable. Each of us will have different opinions on this subject and the questions it raises. Just some that occurred to me watching the video were:

- Would I accept advertising on my favourite websites or television if the adverts aligned with my interests?
- What safeguards are in place to prevent personal information being shared without my permission?

- Could collecting more of my personal data help improve my healthcare?
- Is it possible that my genetic information could be misused to discriminate against me by employers or insurers?

Possibly you thought of other issues.

4.4.5 Data volume versus data quality

The results from any data analysis are only as good as the original data that is being processed. Traditionally, sampling is used to improve the *quality* of the data at the expense of the *volume*. Rather than choosing a random sample from all data, researchers often select the best quality data. They may choose to only use data from the most reliable sources, such as only considering medical results from the most reputable laboratory, or those obtained by the most modern technologies. Similarly, researchers will often deliberately exclude extreme values, concluding they result from experimental error or contamination and are unlikely to be representative of the whole. By contrast, the sheer volume of data involved in big data processing means there is high probability that at least some of the data is of poor quality.

Box 4.6 Monitoring air quality

Suppose a council uses a nitrogen dioxide sensor to monitor a city's air quality every hour. With a single sensor, it is relatively simple to ensure the sensor is producing accurate values; the sensor can be tested not only before being deployed, but also regularly inspected to ensure it is still working correctly. If the council then decides to monitor the city, maintenance of the sensors becomes more challenging – if they deploy one thousand sensors, there is a thousand times as much data to process.

The council might then decide to receive sensor data once every second. Instead of one reading every hour, the system is now handling 3,600,000! This greatly increases the likelihood that some sensor data will be misleading or invalid; a sensor might fail, or readings could be affected by a particularly dirty truck halting under a sensor; or perhaps the sheer volume of sensor data creates congestion on the computer network, meaning some of the data is out of date by the time it comes to be processed.

The possibility that some data might be inaccurate or untrustworthy requires big data applications to perform checks during which misleading values are either discounted or their importance downplayed. This has led to the creation of a fourth 'V':

Veracity: ensuring the correctness and trustworthiness of the data.

Inspired by these Vs, researchers and commentators on big data have started to describe big data using other words beginning with V – including value, variability and visualisation. I won't discuss the five, six or even eight Vs further in this module, as these terms have much less acceptance in the wider community. However, if you do encounter them during your studies, you can learn much more about them using a web search.

Box 4.7 The 'toos'

Some big data researchers prefer to refer to the 'toos', and describe big data as:

- containing **too much** data for traditional databases
- being composed of data that is **too complex** for conventional categorisation (structuring)
- having **too many** updates to the data.

Activity 4.16 (exploratory)

Earlier, I discussed opinion polls as an example of sampling. Perhaps unsurprisingly, big data is now being used by political campaigners to identify individual voters so that they can receive precisely targeted campaign messages likely to swing their opinion. The 2016 United States presidential campaign between Donald Trump and Hillary Clinton was the first to use these techniques on a widespread basis. On the module website there is a video by Channel 4 News which gives some idea of how big data was used. This video shows all three Vs:

1. large **volumes** of data: not only were very large numbers of voters being processed, but large amounts of data about each voter
2. **velocity** was also important: the campaign headquarters were constantly being updated with voter opinions
3. **variety** of data was also present: as well as straightforward opinion polling, the campaigns were using data from social media and credit card purchases to target their message.

4.4.6 Conclusion

Big data is a general term used to describe the processing of data ill-suited to conventional databases. We saw that not only the amount of data being created and stored is increasing, but also the variety of types of data and the speed at which it must be processed to be useful. These changing requirements are known as the three Vs which is a useful tool for considering if a set of data can be considered as ‘big data’.

We also briefly explored how big data techniques allow us to examine data in new ways when previously we were restricted to working with relatively small samples of the entire data set which could lead to serious errors – such as in the forecasting of political elections.

In the next section we will examine how computers process big data to make it useful.

Processing big data

4.5

Big data requires colossal amounts of computing power – not necessarily using a single, ultra-fast supercomputer, but often by assembling more modestly powered computers, called **nodes**, into a **cluster**. Clusters are usually assembled from thousands of identical nodes mounted into racks, each of which consists of one or more CPUs, RAM and a computer network connection. Nodes can have their own hard disks, but many do not, instead storing data on fast disk drives accessed across a computer network.

4.5.1 Cluster computing

Individual nodes are connected using a very-high-speed computer network and overseen through a centralised managing computer responsible for allocating tasks to nodes. Figure 4.8 is a schematic diagram of a simple cluster. The Master computer is where the cluster's operators control the cluster. It is connected using a network cable through a device known as a switch to every node in the cluster. The nodes themselves are connected by the network to large amounts of disk storage for holding data. The disks are also connected to the master computer, allowing the users to access processed data. The master computer may also be connected to the internet, allowing people to use the cluster or view data from anywhere in the world.

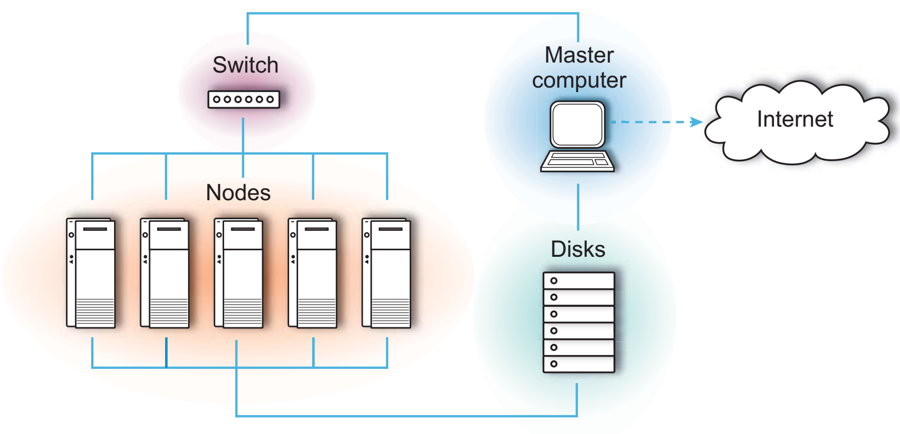


Figure 4.8 A schematic diagram of a simple cluster

Figure 4.9 shows a relatively small cluster computer at the Open University which is used to process scientific data and a close-up of an individual node. The node is designed around a 32 core Intel Xeon processor with 512 GB of RAM. It is connected to the remainder of the cluster by a 40 Gbps network connection.

(a) (b)

Figure 4.9 (a) A small cluster computer with racks of hard disks closest to the camera followed by the 18 cluster nodes themselves; (b) a close-up of an individual node from this cluster (with thanks to Allan Thrower for giving us access to the cluster)

Clusters are a relatively cheap and reliable way of processing large amounts of data.

- Individual nodes are relatively cheap. Whilst most clusters are built from dedicated off-the-shelf nodes, some have been built using consumer PCs or even PlayStation and Xbox video game consoles.
- More nodes can be added to the cluster or old nodes can be replaced with more powerful computers as demand increases.
- Since data processing is distributed amongst many nodes, clusters are highly resilient to failure of one or more nodes; indeed, clusters will continue working whilst individual nodes are replaced or repaired.

Box 4.8 Clusters and grid computing

Grid computers are closely related to clusters and work in much the same way. Grids differ from clusters in that:

- they are often made up from many different types of computer all working on the same task, rather than being built from identical nodes
- they need not belong to the same computer network; instead grid nodes are often distributed across the internet.

Clusters and grids are examples of **distributed computing**. Data processing is distributed amongst the nodes, with every node simultaneously working on a tiny part of the entire problem.

Box 4.9 Distributed computing and parallel processing

Sometimes you will encounter the term ‘parallel processing’, which is a form of distributed computing. In parallel processing, the task is distributed amongst many nodes that share memory. In cluster computing, each node has its own memory and can only communicate with other nodes using a network.

4.5.2 Big data software

Many big data tasks are only made possible by clusters, and this has required the development of specialised software. One of the most influential programs is called **MapReduce**, originally developed by Google. The fine detail of MapReduce is beyond the scope of this module, but its basic workings are reasonably easy to understand.

MapReduce is composed of three software components:

1. a **map** procedure performs a task, such as sorting data
2. a **reduce** procedure summarises the outputs from the mapping
3. meanwhile, the **framework** divides data, sending chunks to each node and receives their results. The framework is responsible for ensuring work is shared fairly between nodes as well as detecting faults with nodes or communication links.

Google formerly used MapReduce to implement PageRank – the method by which the Google web search engine stores and ranks web pages so that the most useful results tend to appear near the top of the first page. PageRank determines the usefulness of a web page based on the number of other pages that link to it. Web pages are catalogued using Googlebot ‘crawler’ programs, which visit billions of websites every day carefully, recording the destinations of *every* link on *every* page. The Googlebots produce huge blocks of data which must be processed to calculate the popularity of every link.

Figure 4.10 shows this process. Blocks of Googlebot data arrive on the left and are broken into equally-sized chunks. The framework then distributes multiple copies of every block amongst nodes in the cluster. More than one copy of each block is processed to prevent data corruption or data loss if an individual node failed whilst processing its block.

Mapping

Once every node receives data, the framework orders them to run identical versions of the map procedure on their data (shown in blue in Figure 4.10). In the case of indexing web pages, each node’s map procedure compiles long lists of every link on every page in the data block.

Input data

Source page	Destination of link
Page 1	Page 2
Page 1	Page 3
Page 2	Page 3
Page 2	Page 4
Page 3	Page 2
Page 3	Page 5
Page 3	Page 4
Page 4	Page 2
Page 4	Page 5
Page 5	Page 6
Page 6	Page 1
Page 6	Page 2

Node 1

Source page	Destination of link
Page 1	Page 2
Page 1	Page 3
Page 2	Page 3
Page 2	Page 4
Page 3	Page 2
Page 3	Page 5

Map

Destination of link	Number of links
Page 2	1
Page 3	1
Page 3	1
Page 4	1
Page 2	1
Page 5	1

Node 2

Source page	Destination of link
Page 3	Page 4
Page 4	Page 2
Page 4	Page 5
Page 5	Page 6
Page 6	Page 1
Page 6	Page 2

Map

Destination of link	Number of links
Page 4	1
Page 2	1
Page 5	1
Page 6	1
Page 1	1
Page 2	1

Shuffle

Destination of link	List of links
Page 1	1
Page 2	1111
Page 3	11
Page 4	11
Page 5	11
Page 6	1

Reduce

Destination of link	Number of links
Page 1	1
Page 2	**4**
Page 3	2
Page 4	2
Page 5	2
Page 6	1

Figure 4.10 A schematic diagram showing how data received from Googlebots is broken into chunks and then processed by the various stages of the MapReduce software to produce a final list of pages and their popularities which can be used by the Google web search engine.

Since every node runs its own copy of the map procedure on its own data, the cluster maps pages in parallel, producing extremely high performance. At the end of mapping, every node holds a long list of intermediate results, but the data needs further processing.

Shuffling

The framework orders every node to sort its list of links into the alphabetical order of the destination page's address. Once every node completes shuffling, the results are combined into a single collated list (green). This new list has one entry for every destination page in the original Googlebot data. Each entry contains information about every page linking to that destination.

Reducing

The volume of data in the shuffled list still exceeds that which humans can handle. Reduction is responsible for summarising the collated list into a more manageable form. PageRank's reduction calculates the total number of links pointing to a web page (green). This reduced list (shown in orange) is then passed to Google's web search engine so that pages with the highest number of links appear first in relevant search results.

Activity 4.17 (exploratory)

For this activity, a short narrated animation is provided on the module website. This animation has been created to show how Google summarised the web using MapReduce to produce data for its search engine. Although Google no longer uses MapReduce for this purpose, it is still used, not only by Google, but also by many other major companies and science projects to make sense of big data.

Although Google now uses alternative approaches to provide data for its search engine, MapReduce lives on as part of Apache Hadoop – a free software package that allows anyone to run their own cluster. Hadoop (see Figure 4.11) is used to provide data storage on the internet (so-called cloud storage) including Amazon's Simple Storage Service (S3) – which powers Tumblr, Pinterest and Reddit – as well as services, such as Amazon Elastic Cloud Computer (EC2) and Microsoft Azure, which allow users to 'hire' computer time to run programs that would be infeasible on their own machines.

Figure 4.11 The Hadoop logo. The project author, Doug Cutting, named the program after a cuddly yellow toy elephant belonging to his son.

4.5.3 Practising for your TMA

For TM111, and throughout your studies, you will be expected to write pieces of text as part of your TMAs, and you will often need to search for information sources as part of this writing task.

This section gives you the opportunity to work through a sample TMA question, whilst talking you through the important stages involved. Activities 4.18, 4.19 and 4.20 are linked activities and will give you practice in locating information from different sources and using it to write an answer to a sample TMA question. You will need to go to the module website for these activities.

Identifying key points

The first stage for a writing question is to identify key points that you need to include, or questions that your written document needs to answer.

Activity 4.18 (exploratory)

In this activity you will locate an article on a particular topic and make notes answering some questions in relation to this. See the module website for the details of this activity.

Study note 4.1 Don't restrict your research to Wikipedia!

Wikipedia is often a good place to begin researching a topic. Many of its articles are of high quality and are up to date. However, because Wikipedia is created by independent contributors, articles can be poorly researched, incomplete or biased. The latter problem is particularly common in controversial topics – especially political or ethical debates. In such cases, you must make a judgement on how much credibility you can assign to the article.

For assessment questions asking you to research a topic, it can be helpful to *start* with Wikipedia, but you should not limit your searches to that site. Instead, after reading a Wikipedia article, use the links in the 'References' section near the bottom of the page to read related pages, as well as any 'External links' to find material elsewhere on the internet. Following initial research on Wikipedia, you can perform further internet and library searches about the topic to ensure you have a comprehensive understanding.

Writing a document

The next stage of writing is turning the notes into a complete document. It is a good idea to use word counts as a guide to your progress. If you have finished your document and you are well under the word count, then there is a chance you have either missed a key point or that you have not included sufficient detail.

Activity 4.19 (exploratory)

In this activity you will use your notes from Activity 4.18 to write a short document answering a sample TMA question. See the module website for the details of this activity.

Study note 4.2 Useful writing tips

When writing your TMA answers:

- write complete sentences, rather than just repeating your working notes
- keep to the topic of the question; don't waste words on irrelevant details
- ensure you have explained any technical terms that might not be familiar to your audience
- likewise, if your document includes any acronyms (e.g. GFT), write them out in full the first time you use them (e.g. 'Google Flu Trends (GFT) is an example of a big data project' …)
- make sure you spell-check your document and correct any errors
- include an accurate word count at the end of your document.

Further investigation

You will often need to look through additional articles and sources of information to investigate the topic further.

Activity 4.20 (exploratory)

In this activity you will use other articles to investigate the sample TMA topic further. See the module website for the details of this activity.

4.5.4 Conclusion

This section introduced cluster computers which marshal large numbers of computers working in parallel to process large volumes of data at very high speed, and it showed the workings of the MapReduce program, one of the first big data applications, to produce search engine indexes. MapReduce is now commonly used as Hadoop which is used for big data processing by many large commercial and scientific bodies.

You then had a chance to practise your research and writing skills by working through a sample TMA question.

The section concludes with a short video about the Square Kilometre Array, one of the largest scientific and computational projects ever attempted. This is provided on the module website.

Summary

This part introduced you to some of the technologies used to store, manage and retrieve data. It began by showing there is a difference between data and information, and the need for data to be processed before it can be turned into useful information.

The first key technology introduced in this part was the database: a method of storing large amounts of data in records made up from pre-defined fields stored in tables. We interact with databases by issuing queries, which are instructions written in specialised query languages. A query attempts to match some or all the data in the database and can be used to retrieve data, as well as add new data and update existing data.

The simplest type of database is the flat database, but – as we saw – flat databases become increasingly inefficient as the database grows and the data it contains changes. Some of these inefficiencies include wasting space and the possibility of inconsistencies within the data.

The solution lies in relational databases, which divide data into several entities which have relationships to one another. These entities are identified and extracted from the data using a process called normalisation. Relational databases are immensely powerful tools, capable of being expanded almost indefinitely and with none of the inefficiencies of flat databases. Indeed, relational databases underpin most of our online society.

The final section of this part was concerned with so-called big data. This is a new approach to data processing which is useful where there is a need to process very large **volumes** of rapidly changing (high **velocity**) highly **variable** data. These characteristics are together known as 'the three Vs'.

Big data is often processed using cluster computers, so we spent some time exploring the workings of clusters and the software they run. In this case, we investigated the MapReduce program originally developed by Google to populate their web search engine.

Now that you have studied this part, you will be able to:

- describe the difference between flat and relational databases using appropriate terminology
- understand key aspects of big data and how it differs from previous database applications
- describe the basic operations of a big data application such as MapReduce
- explore applications for big data.

You will have gained practical experience of:

- interacting with a simple database using queries
- performing a simple normalisation of an existing database to reduce duplicated data and the scope for error
- researching a subject using online resources to answer technical questions
- performing independent research into a topic related to big data.

Answers to self-assessment activities

Activity 4.3

The fields are *Name*, *Date_of_Birth*, *Address*, *Postcode*, *Course_1*, *Course_1_Weeks*, *Course_2*, *Course_2_Weeks*, *Course_3* and *Course_3_Weeks*.

Activity 4.4

Field	**Data type**
Name	Text
Date_of_birth	Date
Address	Text
Postcode	Text
Course_1	Text
Course_1_Weeks	(alternatively – integer or positive integer)
Course_2	Text
Course_2_Weeks	(alternatively – integer or positive integer)
Course_3	Text
Course_3_Weeks	(alternatively – integer or positive integer)

Activity 4.6

When the same data is recorded in several places, there is always a possibility of clerical errors leading to inconsistencies – for instance, the fact that 'Oil painting' taught over fifteen sessions is recorded in three places in the table.

Name	*Date_of_Birth*	*Address*	*Postcode*	*Course_1*	*Course_1_Weeks*	*Course_2*	*Course_2_Weeks*	*Course_3*	*Course_3_Weeks*
Lodhi, Mona	25/05/1979	22 The Grove, Newport	AB12 3CD	Yoga	10	Holiday Spanish	15		
Jones, Bobby	30/11/1990	40 Eldon Court, Hampton	AB6 4PQ	IT for all	5	Oil painting	15		
Cherry, Colin	14/12/1968	59 Acacia Avenue, Brompton	AB2 12ZY	Digital photography	10	Sewing	15		
Cherry, Colin	03/01/1999	8 Mount Pleasant, Greenhill	AB7 4PP	Creative writing	20				
Edwards, Delia	17/08/1987	40 Eldon Court, Hampton	AB6 4PQ	Digital photography	10	Oil painting	15	Ballroom dancing	30
Roberts, Albert	08/09/1952	2 High Street, Stratford	AB12 7UB	Drawing	15				
Singh, Sara	21/10/1977	7 Marina View, Sutton	AB14 8WQ	Woodwork	15				
Chang, Patrick	17/08/1987	21 Green Lane, Newport	AB12 9TU	Everyday maths	20				
Evans, Mary	08/09/1937	13 The Limes, Leighton	AB6 7LR	Oil painting	15				

Activity 4.7

(a) Our *Student* entity has the attributes of *Name*, *Date_of_Birth*, *Address* and *Postcode*. In a properly constructed table there is one field per attribute.

(b) The values are:

- *Name*: Albert Roberts
- *Date_of_Birth*: 08/09/1952
- *Address*: 2 High Street, Stratford
- *Postcode*: AB12 7UB

Activity 4.8

No. There are no attributes of *Postcode*; it is just a single field. Apart from the postcode itself, there is no descriptive information, or label or other qualifying information about postcodes.

Activity 4.9

A key must be unique. Even amongst this small number of students, names are not unique – there are two Colin Cherrys at the college. Delia Edwards and Patrick Chang share a birth date, so *Date_of_Birth* is not a suitable key. Nor can we use *Address* or *Postcode*, since Bob Jones and Delia Edwards share an address. Therefore, we need to create a new key field.

Activity 4.10

(a) If the course is never repeated, then we can safely use the course code as a key, because the code identifies something unique.

(b) If courses are repeated annually, there is a problem with using the course code as a key because it no longer represents something unique. For instance, students enrolling for Yoga in successive years are enrolling for different instances of the course. Each instance of the course would need its own record in the database.

(c) If courses are run twice or more per year, there is a problem with using the course code as a key for the same reasons as in (b). That is, each presentation of the course is a new instance (assuming the database has a lifetime of a year or more), and so the course code does not represent something unique.

The problems raised in (b) and (c) could be overcome by erasing the database either at the end of each year or presentation. The course codes would therefore only ever represent one year's courses. This is not particularly efficient and we would lose information relating to students studying across multiple years or presentations. The OU addresses the problem of multiple presentations by adding a year number and letter to the module code for each presentation: the letter uniquely identifies a month and year, so modules beginning in January 2018 are numbered 18A, those in February 2018 are 18B, and so on.

We don't need to solve this problem in our example, but a database for a real college would need to consider this issue before creating its new database.

Activity 4.11

(a) Students studying multiple courses are those whose *Student_Number* appears in more than one record in Student_Allocation_Table. So, Student 000001 (Mona Lodhi) is studying courses ACT01 ('Yoga') and LNG01 ('Holiday Spanish'). Others are Bob Jones studying 'IT for all' and 'Oil painting', Colin Cherry studying 'Digital photography' and 'Sewing'; and Delia Edwards studying 'Digital photography', 'Oil painting' and 'Ballroom dancing'.

(b) Courses being taken by more than one student have their *Course_Code* in more than one record in Student_Allocation_Table. Two courses are being studied by more than one person. CRT01 ('Digital photography' by students 00003 (Colin Cherry) and 00005 (Delia Edwards); and CRT02 ('Oil painting') by students 00002 (Bob Jones), 00005 (Delia Edwards) and 00009 (Mary Evans).

Activity 4.14

(a) I felt Google Maps showed *volume* and *velocity*. The volume is the number of cars being tracked, whilst velocity is the need to acquire and process traffic data as quickly as possible so that drivers can be warned of current conditions on their route – an hour-old map of congestion might not be valid.

(b) Volume. Yes – YouTube is processing a huge amount of data and the volume is increasing. Velocity. Again, yes – YouTube must make video available as soon as possible to satisfy users, remain competitive against other video sites and obtain advertising income.

References

Arthur, C. (2014) ‘Google Flu Trends is no longer good at predicting flu, scientists find’, *Guardian,* 27 March [Online]. Available at https://www.theguardian.com/technology/2014/mar/27/google-flu-trends-predicting-flu (Accessed 22 March 2017).

Lazer, D. and Kennedy, R. (2015) ‘What we can learn from the epic failure of Google Flu Trends’, *WIRED*, 1 October [Online]. Available at https://www.wired.com/2015/10/can-learn-epic-failure-google-flu-trends (Accessed 22 March 2017).

Lines, N. (2014) ‘What is SEO and how can it help my website’s Google visibility?’, *Guardian*, 7 July [Online]. Available at https://www.theguardian.com/small-business-network/2014/jan/16/what-is-seo-how-website-google-visibility (Accessed 22 March 2017).

van Deursen, A. and van Dijk, J. (2009) ‘Using the internet: skill related problems in users’ online behavior’, *Interacting with Computers*, vol. 21, no. 5−6, December, pp. 393–402 [Online]. Available at http://dl.acm.org/citation.cfm?id=1640752 (Accessed 22 March 2017).

Acknowledgements

Grateful acknowledgement is made to the following sources.

Figure 4.1. Sabre

Figure 4.2. IBM

Figure 4.3. NASA

Figure 4.6. Impru20/https://commons.wikimedia.org/wiki/File:
UK_opinion_polling_2010-2015.png. This file is licensed under the Creative Commons Attribution-Share Alike Licence http://creativecommons.org/licenses/by-sa/3.0/

Figure 4.7. www.statista.com

Every effort has been made to contact copyright holders. If any have been inadvertently overlooked, the publishers will be pleased to make the necessary arrangements at the first opportunity.

Part 5 Weaving the web

by Chris Bissell and Karen Kear

Contents

Introduction

This Part of TM111 is very hands-on, and there are lots of activities to carry out, so you'll need to spend quite a bit of time at your computer and/or online. There is a lot of new terminology, which you can look up in the glossary.

Later in this part I will introduce you to two ways of creating a web page. One involves writing simple code yourself, using a text editor; the other uses Google Sites, which will be used for the major activity in this part. However, before you can understand how information is created for the web, you need to know a little about how the web itself works.

Precisely how long this part will take to study will depend on your prior knowledge. I suggest you allow at least 10 hours, and longer if it is all new to you.

5.1 The web and markup

The web is a **client–server system**. You, the user, sit at your computer and want to get some information from somewhere on the web. Your **browser** sends a request for the information to the computer that holds it. In the context of this interaction, the computer that provides the information is termed a **server** (or more specifically a web server) and your own browser program is termed a **client.** The server finds the requested information on its hard disk and sends it back to the client. The browser then displays the information and you read it, listen to it, watch it, or whatever else you do. The way the browser displays the information is the result of **markup** of the original file – that is, adding special control 'tags' to define such effects as font style, links to other web pages, and so on. You'll learn about the details of markups and tagging in later sections.

5.1.1 Addresses on the web

Each resource on the web is given a unique name, which is made up of several parts. Let's consider a name such as

http://www.w3.org/TR/html4/about.html

This is the name of a web page that contains information on the definition of HTML4, which stands for hypertext markup language version 4 (and which you will learn more about in Section 5.2). The whole name is known as a **uniform resource locator (URL)**. A URL can be split into three parts, as shown in Figure 5.1.

$$\underbrace{\text{http:}}_{\text{scheme}}\underbrace{\text{//www.w3.org/}}_{\text{host}}\underbrace{\text{TR/html4/about.html}}_{\text{path}}$$

Figure 5.1 URL breakdown

- The **scheme name** defines how the client browser should form its requests to access the resource server. For web pages, the scheme 'http', which represents the **hypertext transfer protocol (HTTP)**, is used. You might sometimes see 'https' instead; this represents **HTTPS**, a secure version of HTTP in which all traffic between the client and the server is encrypted so that confidential information such as your credit card details can't be intercepted and read.
- The **host name** is the name of the server that holds the web page. Many computers that serve web pages have names that begin with 'www', but that's entirely optional.
- The **path name** is the name of the file on the server that contains the web page you're after. Typically this file will reside inside a 'nest' of directories (folders), with slashes being used to separate the names of the directories.

So in Figure 5.1, the URL is for a file called `about.html` in a subdirectory called `html4` in a directory called `TR` on a computer called www.w3.org, and the file is to be accessed using the HTTP protocol. Notice that there is a mix of upper-case and lower-case letters: in a URL, upper-case and lower-case are not equivalent, so care is needed.

You will also come across the term **domain name** often used rather loosely as an equivalent to host name. The difference is not really important for the purposes of TM111, but a useful way of thinking about it is that the domain name is ‘top level’, such as wikipedia.org or open.ac.uk, while the various associated ‘lower level’ servers have specific host names, such as en.wikipedia.org or oro.open.ac.uk (the link to the OU’s list of research publications). If you are interested, enter and compare what you see in a browser using (i) wikipedia.org and en.wikipedia.org, and (ii) open.ac.uk and oro.open.ac.uk.

Activity 5.1 (exploratory)

You will have noted that the United Kingdom’s top-level country domain name is *.uk*. Do you recognise, or can you guess, the countries referred to by the following domain names in order to complete the table? They are based on the international standard ISO 3166-1 alpha-2. ISO refers to the **International Organization for Standardization**, which develops and publishes standards on a range of materials, products, processes and services. Adherence to standards improves safety, reliability and quality, and ensures consistency between manufacturers, providers and developers.

Domain	Country
.at	
.au	
.be	
.br	
.ch	
.cn	
.de	
.dk	
.es	
.eu	
.hu	
.ie	
.jp	
.ru	
.za	

Comment

Domain	Country
.at	Austria
.au	Australia
.be	Belgium
.br	Brazil
.ch	Switzerland (*Confoederatio Helvetica*)
.cn	Canada
.de	Germany (*Deutschland*)
.dk	Denmark
.es	Spain (*España*)
.eu	available for organisations in, and residents of, EU member states
.hu	Hungary
.ie	Ireland
.jp	Japan
.ru	Russia
.za	South Africa (*Zuid Afrika*)

Looking at the breakdown of a URL provides a way to investigate, and possibly correct, links that used to work but have 'decayed' because content has been changed or taken down, or servers have become unavailable. For example, if the filename were changed from `about.html` to `about.htm`, then the link in Figure 5.1 may no longer work, and anyone visiting it would see an error page, known as a '404' from the error code used. If that happens to you, try deleting a few characters from the link in your browser. A good starting point is to remove everything after the last '.' and try to load the page again.

Activity 5.2 (exploratory)

Start by entering

```
http://www.w3.org/TR/html4/about.html
```

into the 'Search or enter address' box (or the equivalent) in your browser. This is a page introducing version 4 of the hypertext markup language (HTML).

In the version of Firefox that I use, the search/address box is located near the top of a web page and looks like this:

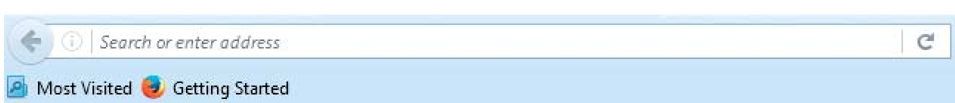


Figure 5.2 Screenshot of Firefox's 'Search or enter address' box

Now delete '.html' from the URL so that it becomes

```
http://www.w3.org/TR/html4/about
```

and access the site. You should see that it has exactly the same effect as

```
http://www.w3.org/TR/html4/about.html
```

– or, at least, it did when I tried it.

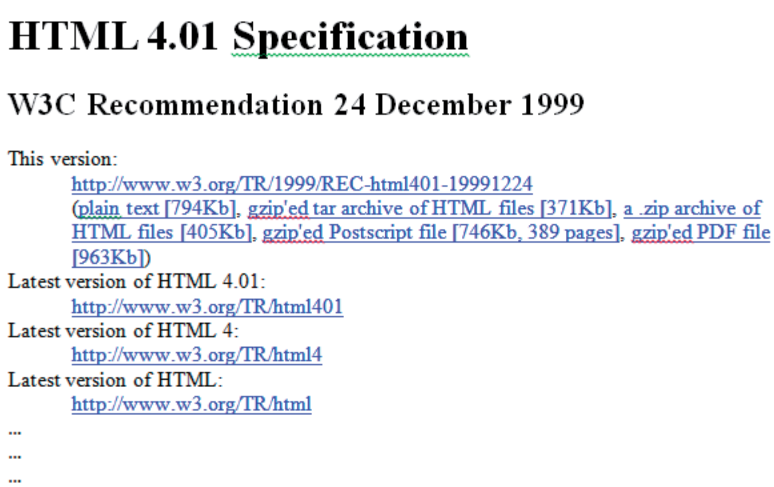
Now delete some more characters to give

```
http://www.w3.org/TR/html4/
```

and access the site.

Comment

When I tried this in November 2016, in the final step of this activity I saw a page beginning with:



HTML 4.01 Specification

W3C Recommendation 24 December 1999

This version:
http://www.w3.org/TR/1999/REC-html401-19991224
(plain text [794Kb], gzip'ed tar archive of HTML files [371Kb], a .zip archive of HTML files [405Kb], gzip'ed Postscript file [746Kb, 389 pages], gzip'ed PDF file [963Kb])
Latest version of HTML 4.01:
http://www.w3.org/TR/html401
Latest version of HTML 4:
http://www.w3.org/TR/html4
Latest version of HTML:
http://www.w3.org/TR/html
...
...
...

Figure 5.3 Screenshot of part of W3C's HTML 4.01 Specification page

Now scroll down the page. You should see a table of contents and clicking on the link in item 1 takes you back to the page accessed by http://www.w3.org/TR/html4/about. In other words, by stripping off the 'about' section you have gone 'up' a level.

Backtracking through URLs can occasionally be useful – for example, to help you find a page when you discover that a link no longer works – although it's not recommended as something to do routinely.

Not all sites will let you see a list of folder pages in this way. Sometimes when trying to access a web page via its URL you may get an error message instead, and sometimes you may get a default page called 'index.html' or 'default.html' or similar. If you see such a page, it means that the server has been configured to display it to anyone trying to look at the contents of the folder.

A related point worth noting is that the server may have been configured not to show the full path name. Often something as simple as bt.com, bl.uk (The British Library) or open.ac.uk can be used.

5.1.2 Creating HTML

So how might you go about writing a web page?

The content of a web page is put together using **hypertext markup language (HTML)**, the programming language of the web, which your browser understands. A browser needs a set of codes to identify the various parts of the web page: the headings, lists, paragraphs and so on. These codes are called **tags** in HTML and they make it possible for a web page to be displayed in a browser. HTML tags are standardised so that they can be interpreted by different browsers.

HTML was designed to be easy to use. The basic tags are very simple, and any text editor (such as the Notepad and TextEdit applications provided with Windows and Mac OS X, respectively) can be used to create HTML documents. All you have to do is learn what the basic tags are and where to put them.

However, as the web has become more popular, people have created more elaborate web pages, with complex layouts and more interactivity. To cope with this complexity, two strategies were developed to help HTML users.

- There are many editing applications that are specifically designed to work with HTML. An **HTML editor** usually has the functionality of a word processor – so, for example, you might simply click on a button to make text appear bold – but behind the scenes it is automatically inserting the HTML tags necessary for a browser to display your **styled** text.
- Many ordinary word processors offer a command such as 'Save as HTML', which converts a word-processed document into HTML by automatically inserting the necessary tags (but this method can sometimes create overly complex HTML).

So is there still a need to learn HTML? Strictly speaking, if you are interested only in creating a simple web page, then no! With the more advanced word processors or with an HTML editor, you can create web pages without ever needing to see an HTML tag. However, there are various reasons why it is still useful to have a basic knowledge of HTML if you want to create web pages.

Sometimes you will find that the special tools in a word processor or HTML editor do not give you quite what you expected. Then you may need to look at the HTML itself to see what has gone wrong. Or you may find a web page that has an interesting feature and wonder how it is done. Then you will need to look at the HTML involved. You can do this using the ‘View Source’ or equivalent command in your browser, to see how it has been accomplished and to learn from what others have done. And, finally, anyone claiming to be a computing or IT specialist should know something about markup languages as part of their general ‘toolbox’ – not to mention the advantage of such knowledge for employability.

5.1.3 Conclusion

In this section you were introduced to:

- client–server systems
- URLs and their structure
- HTTP and HTTPS protocols
- the HTML markup language.

Now it’s time for you to try creating a web page of your own.

5.2 Beginning HTML

In essence, HTML is a way of **styling** a document (different font sizes and other typographical features) but, more importantly, a way of including automatic links to other documents, both local and held on web servers anywhere in the world.

Almost anyone can put material online by creating a web page – but in order to make your own material available on the web, you first need to have somewhere to put it. To use the jargon, you need some space on a web server.

Individuals do not normally have their own web servers residing on a computer of their own. However, it's easier and cheaper than you might think to purchase a personal domain name hosted by someone else, so some people do just that. There are also many providers who offer free personal web page hosting.

One such provider of personal website hosting is Google Sites. This is the host we shall use in TM111. Later you'll use the HTML editor built into Google Sites. You will then look at what's happening under the surface of the page, by using the Google Sites editor's facility to view the HTML that it generates. I advise you to keep a regular eye on the module website for additional resources, such as chunks of code you can modify in some of the activities, and for any updating messages about Google Sites.

5.2.1 Content and presentation

First, think about how you are viewing what you are reading now. There is text of various sizes. There are some graphical features, such as images. Different words are emphasised in different ways.

Furthermore, you may be looking at it on a printed paper copy or on a computer screen. You may be reading it on your mobile phone or a tablet. You may have your browser or word processor application settings changed to increase the contrast or size of font because your eyesight isn't as good as it was. Perhaps you're using an application to enlarge a portion of the screen, or using a screen reader and listening to the page rather than viewing it. There are lots of different ways of accessing the page, each of which has a different way of delivering the content. Yet fundamentally, despite all the differences, you're still reading the same page. You're gaining the same information from it, and you're (hopefully) thinking about the same things whilst doing so. In other words, the same *content* is being *presented* in different ways. This is the key idea I want to develop now.

Different browsers and devices

HTML was designed to be used on any computer, but it wasn't designed to give precise styling instructions to the browser that displays the HTML document. Instead, HTML provides broad guidance on the relative sizes of text, the alignment of lines of text and the placement of graphics. Thus there are sometimes variations in how different browsers display an HTML document on the screen. This is especially true for non-PC browsers such as those used on phones, tablets, and other handheld devices. These devices often have much smaller screens that can't display all the **elements** (that is, what is generated *after* the interpretation of tags) in the code as designed for laptops and desktops.

Activity 5.3 (self-assessment)

(a) What does HTML stand for?

(b) HTML is increasingly device-independent. Use your web searching skills to find out what 'device-independent' means and what advantages it gives.

5.2.2 Improving readability

The previous section introduced the idea that HTML allows you to give a general indication of how your text should be styled, with the exact appearance depending on the particular browser and device. But why would you want to style your text at all?

For most published materials (including those on the web) the author needs to bear in mind that he or she is addressing a particular audience with particular needs. For example, the audience of a magazine article will probably be reading for pleasure, possibly with distractions in the background. Or the audience may be, as *you* are here, trying to learn from the material being read. In such cases, the author needs to engage you by providing helpful visual clues that emphasise important points, highlight interesting but less important points, and provide a clear structure that aids your learning.

5.2.3 Marking up for style

An obvious requirement of markup is that it should be able to distinguish styled text from unstyled text.

The '**marks**' of a markup language are part of a computer document; in other words, they must be *embedded* within the electronic text. But how are they to be distinguished from the text itself? If the word 'italic' is used as a mark and placed next to a word that is to be italicised, how does the browser know that 'italic' is not just part of the text?

Electronic markup uses an opening tag at the start of the area to be styled and a closing tag to indicate where the styling ends. In computing, this concept of marking a start and end point is referred to as **delimiting**. Pairs of tags delimit areas of a document that are subject to special styling, as you will see in the examples below.

Angle brackets (< , >) are a good choice for marking up because they seldom appear in normal text outside mathematics, particularly in the configuration of left bracket, letter, right bracket. In electronic markup these abbreviations are called *tags*. HTML, the hypertext markup language, uses such angle brackets to mark its tags.

5.2.4 Explicit markup: your first home page

Now you've got the idea of using markup to encode the presentation of a document, how do you go about entering both the content of the document and its styling into a computer? In this subsection you will write all the HTML content character by character. This is called **explicit markup**. When you do this, you can see the marks on your computer screen while you are developing a page; it is only after the document passes through some suitable software (such as a browser) that it will be displayed or 'rendered' with the intended appearance.

Let's begin by looking at some simple HTML code. Note that each start tag `<tag>` is closed by the end tag `</tag>`. For example, the tag `<h1>` opens some level-one heading text and `</h1>` closes it. You can often guess the meaning of tags: for example, `<b>` means 'bold text' and `</b>` means 'turn off bold text'.

Some simple HTML code

Read quickly through this short section of HTML code:

```
<h1>This is a level one heading</h1>
<p>This is a paragraph with a break before and after</p>
Here is some <b>bold text</b> and here is some
<i>italic</i> text
<p>another paragraph plus breaks</p>
You can have <b><i>bold and italic together</i></b>
```

Note that `<b>` and `<i>` simply mean 'use bold text' and 'use italic text' and `</b>` and `</i>` mean 'turn the formatting off'. The last line is an example of *nesting*, where one tag is inside another, meaning 'use bold italic text'. In such cases, it is good practice to put the end tags in the reverse order to the start tags.

Activity 5.4 (exploratory)

- Identify the tags in the above code. Make a list of them in alphabetical order, for future additions to your list, with a note of what they do, and keep updating the list as you come across new tags.
- On your computer, open a new folder for your website files. Type all the code shown above (or copy it, if you have it electronically) into a text editor such as Notepad. Save it as a file called `homepage1.txt` (.txt will be the default filename extension, and may not appear on your screen).
- Now save the file as `homepage1.html`. To do this in Notepad you select the 'All Files' option and type in the filename, including the .html extension. Keep the ANSI option if offered a choice. Finally, save it in the same folder as `homepage1.txt`
- Open the HTML file using a browser. With a bit of luck, this should simply require you to double click, or right-click and then select 'open', on the icon of the .html file, but some browsers are not so straightforward, and you may need to consult their 'help' file.

When opened in a browser, you should see:

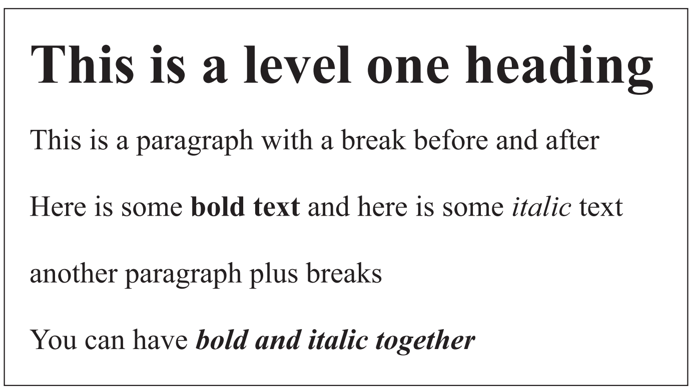


Figure 5.4 Screenshot of the sample HTML file displayed in a browser

If you don't get this, check that each element has the correct opening and closing tags (delimiters) – this is one of the most common errors in writing HTML code.

Activity 5.5 (exploratory)

Now let's make something that looks more like a rudimentary home page. Using the tags in Activity 5.4, plus the one for a level 2 heading `<h2>`, create something relating to yourself, looking like the example in Figure 5.5 when displayed by a browser. Give it the filename 'homepage2.html'. You'll gradually develop your home page in subsequent activities, so make sure that every time you are happy with a new version, you save it with a suitable new name.

If you get something unexpected, check that (i) each tag is delimited, (ii) any nesting is in the correct format and (iii) you have included enough paragraph breaks of the form <p></p> to provide desired spaces between lines. Note that tags do not distinguish between upper and lower case.

Chris Bissell's Home page

I've been teaching and researching at the OU for quite some time, in fact **I'm well into my fourth decade!**

I have contributed to many OU modules over the years, most recently TM355 *Digital Communications*.

My research interests are in engineering modelling, engineering education and the history of technology.

Leisure Interests

I am interested in all types of music, particularly classical, jazz, blues and some country, and I play the piano. I'm also a fan of art, and like to visit galleries and exhibitions.

Figure 5.5 Screenshot of a sample rudimentary home page

5.2.5 Links to other files

So far, so good, but the beauty of HTML is that it allows clickable links to other files and web pages. Such files can be local to your computer, hosted on your own site, or held anywhere on the web.

I'll not say much about including other files by doing your own marking up, except for you to try two examples. You will create other links later when you use an HTML editor.

Adding a link to a local file

It is easy to add a link to a local file, such as an image, held in the same folder as the homepage files. The format for inserting an image file called `picture.jpg`, held in the *same* folder as the HTML file, is

```
<img src="picture.jpg">
```

Note that, unusually, there is no specific end delimiter – simply the paired brackets < … >.

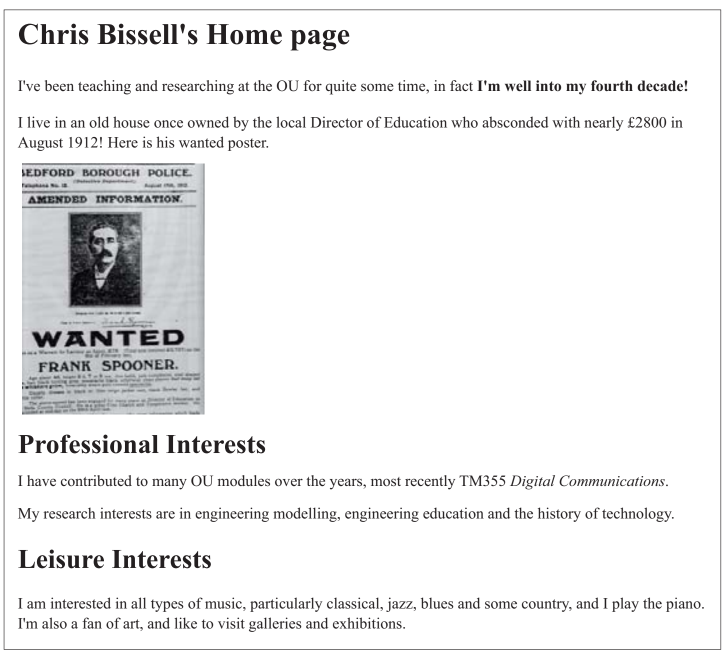
This format will produce a default-sized image. The ‘src’ element stands for ‘source’, and can be extended to point to images on other websites, as you will see later. The code can also be amended to change the picture size, but we’ll leave that for now, so choose a small image displaying on your screen as a maximum of 5 × 5 cm (unless you are competent at re-sizing using a program such as the Windows Paint accessory). If you do not have a suitable image of your own, use that in Figure 5.6.

Figure 5.6 The Open University logo

You can grab the image from an OU website using the Windows Snipping tool or one of the .jpg variants to be found at www.open.ac.uk/about/brand/print/logo. Using the Windows Snipping tool, simply select the portion of the display you want to capture, and save it in a suitable format in the folder you wish.

Now try adding an image into your page. Browsers will read most common image formats such as .jpeg, .png, .gif and others.

Here’s what I came up with.



Chris Bissell's Home page

I've been teaching and researching at the OU for quite some time, in fact **I'm well into my fourth decade!**

I live in an old house once owned by the local Director of Education who absconded with nearly £2800 in August 1912! Here is his wanted poster.

Professional Interests

I have contributed to many OU modules over the years, most recently TM355 *Digital Communications*.

My research interests are in engineering modelling, engineering education and the history of technology.

Leisure Interests

I am interested in all types of music, particularly classical, jazz, blues and some country, and I play the piano. I'm also a fan of art, and like to visit galleries and exhibitions.

Figure 5.7 Home page with image inserted

Including external links

The real power of HTML, though, is when it is used to address an external link. The format uses a tag known as an *anchor,* `<a>`, with a structure of the form:

```
<a href="https://www.youtube.com">YouTube</a>
```

The first part gives the URL of the document you want to address and the second the hypertext that links to it, so the above code will appear in a web page as simply ‘YouTube’. When you click on ‘YouTube’ it will bring up the YouTube website. The linked text ‘YouTube’ is known as a **hyperlink** (**hotlink**).

Now try to add a couple of hyperlinks to your page.

This is the result of the changes I made to include links to an OU site about TM355, and The Courtauld Gallery. (Note that the text linking to the URL can contain other styling, such as italic.)

Professional Interests

I have contributed to many OU modules over the years, most recently TM355 *Digital Communications*.

My research interests are in engineering modelling, engineering education and the history of technology.

Leisure Interests

I am interested in all types of music, particularly classical, jazz, blues and some country, and I play the piano. I'm also a fan of art, and like to visit galleries and exhibitions. One of my favourites is The Courtauld Gallery.

Figure 5.8 Home page with hyperlinks included

5.2.6 Other features of HTML

You will meet many other features of HTML when you use an editor. Some of the most frequent are:

- specifically coding for a given font, size and colour
- using hyperlinks to jump to specific parts within a long document
- producing bullet points using the tags `<ul>` (unordered list) and `<li>` (list item). Add these to your alphabetical list of tags that you should have been updating. (If you haven't, update it now!) The format is

```
<ul>
<li>one item</li>
<li>and another item</li>
<li>a third item</li>
```

and so on until you get to the end of your bulleted list, then finish with:

```
</ul>
```

HTML is also designed to give ‘**graceful degradation**’ – in other words, it will try to deal with certain possible errors in the code without crashing completely. For example, your browser doesn’t mind how HTML tags are written, as long as they are spelled correctly, so it interprets <P> in exactly the same way as <p>. Another important aspect of this is that browsers are designed to ignore tags they don’t recognise – for example, if I accidentally type <hi> instead of <h1> when setting a level 1 heading.

If a page contains incorrect tags then the browser will still try to display the content, though it probably won’t be able to do it in the way you expect. This is important because HTML is continually changing and having more tags added to it – often before browsers support them. As a result, it is very important that browsers display as much as possible of the content of a document, even when they don’t understand all the tags.

5.2.7 Using the Google Sites HTML editor

The following lists the basic Google Sites editorial processes, for ongoing reference as you develop HTML code. You will find a current version of this on the module website, and you might wish to print it out and display it where you use your computer.

1 Select the editor (pencil tool) and you should see the editing menu:

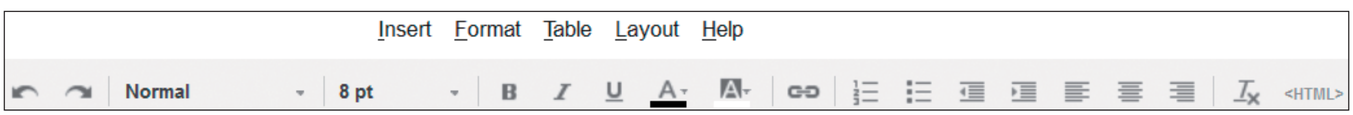


Figure 5.9 The Google Sites editing menu

2 ‘Format’ gives you options such as:

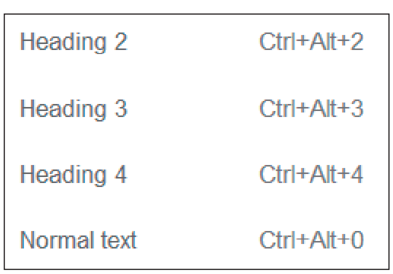


Figure 5.10 Google Sites formatting options

3 ‘Normal’ gives you font options:

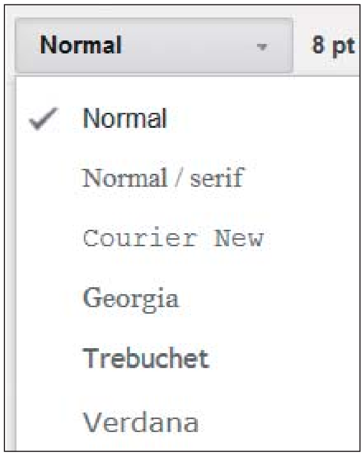


Figure 5.11 Google Sites font dropdown menu

4 ‘Insert’ gives you various options. The most important for simple design are:

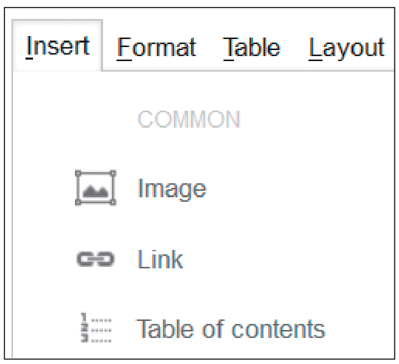


Figure 5.12 The Google Sites ‘Insert’ menu

5 ‘Create Link’ (chain symbol) gives you:

Figure 5.13 Google Sites link options

Highlight the text that will become the link, then use ‘Create Link’. The highlighted text will appear automatically in the ‘Text to display’ box; then select ‘Web Address’ to add the link URL.

6 Upload a local image to the Google server by selecting ‘Insert Image’, and ‘Upload Images’.

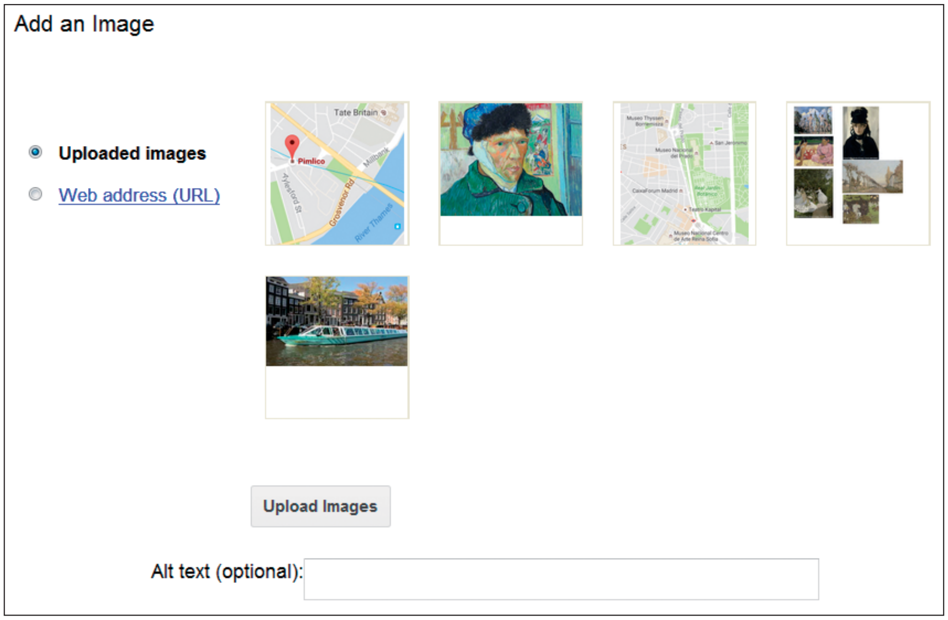


Figure 5.14 Google Sites image uploader

Previous uploads will show as thumbnails. *Alt text* can be used to provide a description for visually impaired users.

7 Directly editing the HTML source: the small symbol `<HTML>` on the menu displays the HTML code in a separate editing box, so you can modify it directly (such as when including internal jumps, as described below).

Activity 5.6 (self-assessment)

(a) What is meant by the term ‘nesting’? Can you nest HTML tags?

(b) Do browsers require all tags to be present in order to work?

(c) Do HTML tags need to be spelled correctly?

5.2.8 Conclusion

This section has introduced HTML and the use of tags in the explicit markup for style in order to generate a simple, personalised home page. You have learnt how to incorporate images and links to external websites into such a page. You have been introduced to the major features of the Google Sites HTML editor, and have seen the menu options used to style text, insert images and links, or view the original HTML code of the page.

5.3 Designing a web page

In this session you will be using the Google Sites editor. However, all the HTML you'll learn about could be explicitly marked up using a text editor, as you did in Section 5.2.

If you look carefully at well-designed web pages, you'll notice they tend to:

- be relatively short, so that too much scrolling is avoided, depending on the device display
- put other information on subsidiary pages, with links connecting the various parts
- not use too many subpages
- use different fonts and colours sparingly, in line with good design practice.

There is one very important point to note here: hypertext doesn't actually allow a reader complete freedom to move around the text! The author of the text will have decided whilst developing the document how it is to be structured, what links are to be included, and where they should point to: the so-called *targets*. Thus the design and structuring stage is a very important part of preparing hypertext.

Study note 5.1

All authors have to learn a discipline of writing that involves, above all, deciding how a document is to be structured. With hypertext, this skill becomes even more important. It is not an easy skill to learn, and many published authors find that they still struggle with the design and structure of each new work. However, they also admit that getting the structure of a document right – be it an article, instructions, a report, a book or a web document – is as much the key to producing a good text as having the correct factual information.

5.3.1 Images from external sources

You have already seen how to link to an image file in the same folder as your HTML document using the `<img>` tag. But you can also insert images located elsewhere on the web, for example:

```
<img src="http://www.open.ac.uk/includes/
oulogo_hor_75x55.gif">
```

Here the `src` attribute indicates where the image file can be found (its source). This particular piece of HTML would insert a small version of the OU logo into the document by directly accessing the OU website.

Now is a good time to update the list of tags you started in Activity 5.4.

5.3.2 Internal links within a page

You can also include hyperlinks to places within the same document. This is most often useful with long pages, where the links are to sections within the page. For instance, look at a typical Wikipedia page (such as the one on hypertext at http://en.wikipedia.org/wiki/Hypertext) and you should see that it has several sections and subsections, with a table of contents at the start of the article. Each line in the table of contents links to the corresponding section in the article.

So how can you make this sort of internal link in your documents? As well as using the `<a>` tag to mark up an external link, as you did in the previous section, you can also use it for an internal link – that is, to designate a particular point in the document as the fixed reference point or anchor for a link elsewhere in the document. To do this, you need to use the `name` attribute of the `<a>` tag.

To do this, read Study note 5.2 but do not try to follow it completely or memorise it. You will put it into practice in the next activity. The most important point to remember is that you have to make the changes in the HTML code by hand.

Study note 5.2

Inserting internal links is straightforward, although it is easy to make errors when you write the HTML code. In my page https://sites.google.com/site/paintingspage/ I put a bulleted list of contents at the beginning. To jump from a bullet point to somewhere further down the page you edit the HTML (in Google Sites, click the symbol on the right of the edit menu) and then modify the format for the bullet to:

```
<li><a href="#London">London</a></li>
```

Note that Sites Help suggests you use a full URL

```
<a href="https://sites.google.com/site/
paintingspage#London">London</a>
```

but this isn't really necessary. Whichever you use, the bulleted word London will appear as a hyperlink.

Then use the format

```
<h2><a name="London">London</a></h2>
```

to indicate the place you want to jump to – in this case, the heading 'London' later in the page.

When you are happy with your modified HTML code, you can click on ‘Update’, and check that the changes appear as you expected on the page.

Activity 5.7 (exploratory)

This final activity is the most substantial part of this section. If you have not used Google Sites or a similar HTML editor, it could take you quite a while.

Please consult the module website for further guidance. If you attempt the activity from scratch, make use of the summary instructions below, or use the guidance on the module website. If you already have experience in HTML, allow 2 hours; if it is all new to you, you might need 4 hours or more, so do it in stages.

When you first start to create a page, Google Sites offers you the choice of *Classic Sites* or *New Sites*. Most people find *Classic* more straightforward, and most of the online help is for *Classic*, rather than *New*. So it is probably best to stick to *Classic*.

Note that Google Sites pages can be made private or publicly accessible. If you wish to limit access to yourself during development, click on the Share button and change the access permissions:

Figure 5.15 Google Sites ‘Share’ button

Create a significant web page for yourself on a subject that interests you. The topic could be about your family, a hobby, a particular interest, a club you’re a member of, and so on. By significant, I mean a page containing potentially useful information to the user, with at least:

- two external links to resources of different types (text, audio or video)
- one bulleted list
- one embedded image
- one jump within the page, and one to a second, related page you designed yourself. (Just create a new Google Sites page and link to it from the first page.)

I decided to add some notes on ‘looking at paintings’ on a second page linked to my main page https://sites.google.com/site/paintingspage/. Bear in mind the following guidelines:

- Start by making a sketch of the general appearance of your page, showing where the external links will appear. Go easy on the variety of fonts and colours. When you use the HTML editor, I suggest you just use a blank page, or one of the standard template options.

- If the page does not render as you expected at various stages of development, you may need to check the source HTML for the typical errors you met earlier (although most of these should be dealt with in the Google editor). Watch out for stray spaces in link URLs, which may render them inaccessible.
- Do not get carried away in adding more and more features, at least not as part of your study time – as with all coding, it can easily get out of hand!
- Try to use the various elements you've already met: different-sized headings; some bullet points; a local image (perhaps a photograph or a sketch of your own); a variety of external links to web pages, videos, audio, etc. Make sure you do not infringe any copyright, although you should be OK to link to reputable sites.
- Put in some internal page links to jump to specific parts of the page, using the instructions in the previous box.
- Use Sites Help if you get into difficulties – it's pretty well written.
- As mentioned above, my main site is page https://sites.google.com/site/paintingspage/, but you do not need to produce anything so detailed.

When you are happy with your page, upload a link to your web page into OpenStudio. See the module website for guidance on how to do this. When enough of your fellow students' pages have been published, choose two or three, and for each, comment on:

- at least *two* particularly good or effective points
- no more than two aspects that could be improved.

As always when commenting on other people's work, be positive and supportive!

5.3.3 Conclusion

In this section you have worked in Google Sites to produce a much more detailed web page than in the previous section, complete with embedded images, internal jumps, and external links to various sources. You should have posted a link to your main page for fellow students to comment on, and commented in return on two or three of theirs.

5.4 HTML5

At the time of writing, the most recent HTML version is known as HTML5, and includes a number of features not present in earlier versions of the language. This section aims to give you a brief introduction. You are not expected to do any original coding using HTML5, but you will be asked to explore the effects of modifying some existing code.

5.4.1 Introduction to HTML5

Here is some **HTML5** source code, taken from a tutorial at http://www.html-5-tutorial.com. You do not need to understand all this code – just try the activities that follow.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>My fifth webpage</title>

</head>

<body>

<div class="outer">
<div class="header">
<h1 class="c">Header</h1>
</div>

<div class="nav">
<h2>Navigation</h2>
<ul>
<li><a href="my-first-webpage.html">My first page</a></
li>
<li><a href="my-second-webpage.html">My second page</
a></li>
<li><a href="my-third-webpage.html">My third page</a></
li>
<li><a href="my-fourth-webpage.html">My fourth page</
a></li>
<li><a href="my-fifth-webpage.html">My fifth page</a></
li>
</ul>

</div>
```

```
<div class="article">
<p>Play with the html and css all you want, I'm going to
replace this page in the next lesson anyway.</p>

<p>As you play with it you will find strange things can
happen or that nothing happens when it should have. Keep
track of dots, brackets, colons and semicolons. That's
often where things go wrong.</p>

<p>You will always have a backup in that you can just
reload my copy off the server, open the source code, copy
and paste into your editor (probably Notepad), and try
again.</p>

<p>The main thing is to not get discouraged. There is a
lot to know and you are just getting started.</p>

</div>

<div class="footer">
<h3 class="c">Footer</h3>
</div>

</div>

</body>

</html>
```

You will find electronic versions of all the code in this section on the module website.

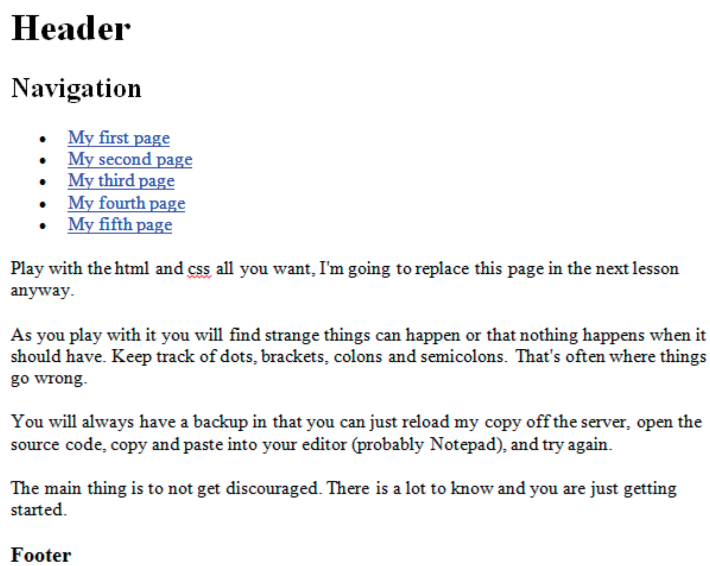
Activity 5.8 (self-assessment)

Which tags do you recognise, and which are new? You do *not* need to understand the precise role of the new tags, or be able to generate code using them. However, you should identify them and be able to infer some of their functions.

Activity 5.9 (exploratory)

Paste the above code into Notepad, and save it as an HTML file as you did in earlier sections. What do you see when opening with a browser? Note that the text in the sample code refers to the HTML5 tutorial, not this module activity.

Comment



Header

Navigation

- My first page
- My second page
- My third page
- My fourth page
- My fifth page

Play with the html and css all you want, I'm going to replace this page in the next lesson anyway.

As you play with it you will find strange things can happen or that nothing happens when it should have. Keep track of dots, brackets, colons and semicolons. That's often where things go wrong.

You will always have a backup in that you can just reload my copy off the server, open the source code, copy and paste into your editor (probably Notepad), and try again.

The main thing is to not get discouraged. There is a lot to know and you are just getting started.

Footer

Figure 5.16 Screenshot of the HTML5 web page

Note that the links are 'dummies', and so do not point to anything, unless you have files with the relevant names. Now, you may think that this page could have been written more simply in the type of HTML you learned in earlier sections. This is true, but we'll look at how to significantly improve the page's presentation in Section 5.5.

5.4.2 Multimedia

HTML5 includes a number of tags designed specifically for embedding multimedia displays, such as video, in the web page itself. If you are interested in a complete list of new tags and other elements, see http://www.w3schools.com/html/html5_new_elements.asp. New media elements include:

Tag	Description
`<audio>`	Defines sound content
`<embed>`	Defines containers for external applications (like plug-ins)
`<source>`	Defines sources for `<video>` and `<audio>`
`<track>`	Defines tracks for `<video>` and `<audio>`
`<video>`	Defines video or movie content
`<iframe>`	Specifies an inline frame

One of the simplest of the HTML5 tags to use is the `<iframe>` tag, which specifies an **inline frame**. Here the term 'frame' is used to mean multimedia that appears directly in the web page, which can thus be run directly. Hence an HTML5 inline frame can be used to easily embed a video or audio clip within the current HTML document. (Note that the <iframe> function was available before HTML5, but was not straightforward to use and its success or otherwise was dependent on precise file format and computer configuration.) In the example that follows, the <iframe> tags embed the Cortauld YouTube video.

```
<!DOCTYPE html>
<html>
<body>

<iframe width="420" height="345" src="https://www.
youtube.com/embed/Qi8oNqJQcWk"></iframe>


</body>
</html>
```

You can add additional text in the usual way, thus:

```
<!DOCTYPE html>
<html>
<body>
<h2> This is the video I used earlier about the Courtauld
Gallery on my <i>Paintings Page</i></h2>

<iframe width="420" height="345" src="https://www.
youtube.com/embed/Qi8oNqJQcWk"></iframe>
<p>
<b>This time it is "embedded" using the HTML5
"<i>iframe</i>" tag. The frame displaying at the moment
shows the presenter in front of the famous <i>A Bar at
the Folies-Bergère</i> by Eduard Manet.</b></P>

</body>
</html>
```

This code, when displayed by a browser, shows the embedded video. In Figure 5.17 I have shown a point part way through the video for illustration. But the clip will start from the beginning (Adam and Eve) when you click on the play button.

This is the video I used earlier about the Courtauld Gallery on my *Paintings Page*

This time it is "embedded" using the HTML5 "*iframe*" tag. The frame displaying at the moment shows the presenter in front of the famous *A Bar at the Folies-Bergère* by Eduard Manet.

Figure 5.17 Screenshot of home page with YouTube video embedded

Audio can be embedded similarly, using the tag `<audio>` linking to an internal or external resource. For example:

```
<h3> Well, I'm becoming a fan of HTML5. Audio is straightforward too, either with a local file</h3><p>

<audio controls>            adds the graphical controls

<source src="openuniversity.mp3" type="audio/mpeg">

</audio>                    file located on my computer in same folder
</p>
<p>
<h3>or an a external link</h3></p>
<p>

<audio controls>

<source src="http://lisztonian.com/music/download/Invention+No+1+in+C+Major-109.mp3" type="audio/mpeg">

</audio>
</p>
```

Figure 5.18 HTML code for a page with embedded audio clips

When saved as an HTML file, this displays as

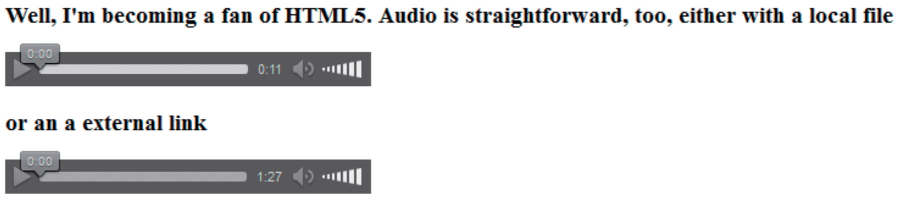


Figure 5.19 Screenshot of a page with embedded audio clips

Activity 5.10 (exploratory)

(a) Save the code for the embedded video as a .html file, open it using a browser, then check that the video plays properly.

(b) Modify the text and the YouTube video resource to something of your own choice.

(c) Create a page with embedded audio, including:

  (i) a link to a local audio file of your own (perhaps from your work with Audacity)

  (ii) a link to an external audio file.

Note that if you want to use the OU theme tune as either internal or external audio, you'll need to locate a copy on the web.

5.4.3 Conclusion

The main part of this section has introduced some of the new features of HTML5, including some tags that did not appear in earlier versions of the standard. You have learnt in particular how to embed a video or audio file into a web page using the `<iframe>` and `<audio>` tags.

5.5 Stylesheets

Well-designed web page presentation is vital for readability and appeal. The appearance of a page is greatly simplified by the use of stylesheets. A **stylesheet**, often known as a **Cascading Stylesheet (CSS)**, is code used to format a page in a desired way: the presentation information goes in special `<style>` tags, often as a separate page of code. Indeed, a CSS can control the layout of multiple web pages all at once and thus save a lot of work. A style can be added to HTML elements in three ways:

1 inline – by using the ‘style attribute’ in HTML elements. A style attribute applies individual formatting to a single element, such as

```
<h1 style="color:blue;text-align:center">
This is a heading</h1>
<p style="color:red">This is a paragraph.</p>
```

which gives

Figure 5.20 Screenshot showing effect of inline styling example

2 internal – by using a `<style>` full stylesheet in the `<head>` section, which operates on the entire page; there is no need to style individual tags as in the previous example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

This gives the display

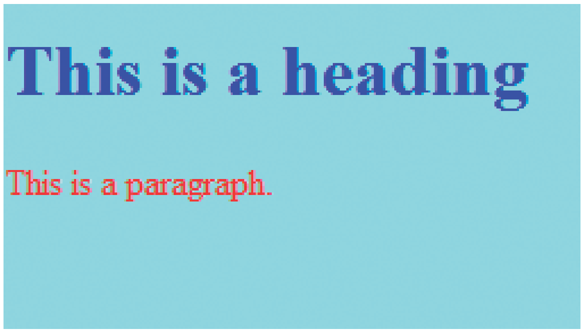


Figure 5.21 Screenshot showing effect of internal styling example

3 external – by using an external CSS file, which can be applied to multiple documents.

The most common way to add a CSS is to keep the style in a separate external CSS file. The browser **rendering** the page knows to use that stylesheet because it's included in a tag in the HTML page's *header* (the information at the top of the page). For instance:

```
<link href="formatting.css" rel="stylesheet"
type="text/css">
```

As you might have guessed, the three elements in this tag refer to an external *text stylesheet* called *formatting*. However, you do not need to understand the details of such a tag, or be able to use them.

Note that in the example above, the colours are specified as words – this limits the options. To increase the colour palette, hexadecimal values are used for providing a huge range of shades. If you have not met **hexadecimal (hex)** before, it is explained in detail in the next subsection. For the time being, just accept that it is a way of writing long binary numbers in a much more concise form. You can explore HTML colour codes at: http://www.w3schools.com/colors/colors_picker.asp

Activity 5.11 (self-assessment)

Save this code as a .html file and notice the new colours defined by the hex values.

```
<!DOCTYPE html>
<html>
<head>
<style>
body { background-color: #ffbf80;}
h1 { color: #006600;}
p { color: #6600cc;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

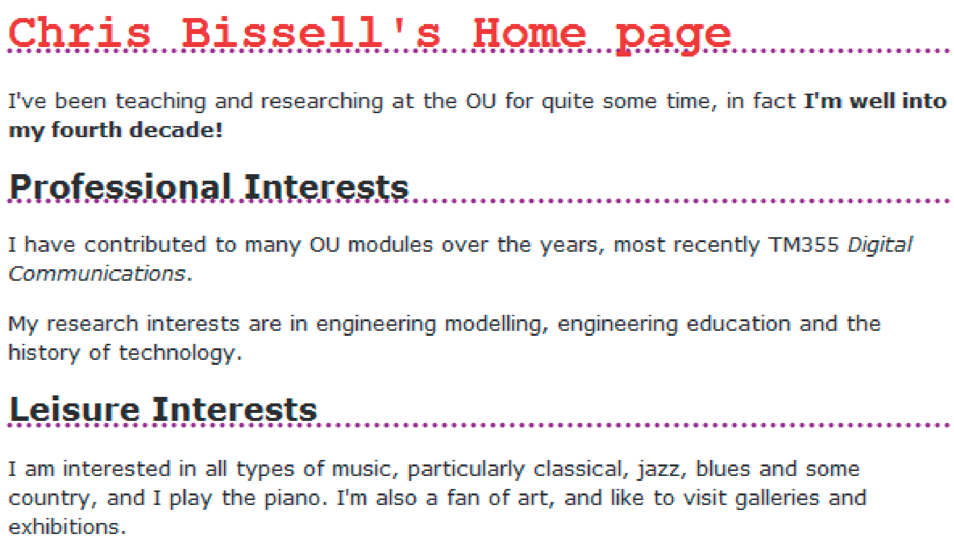
Before leaving the topic of CSS, have a look at the following stylesheet. You don't need to understand the fine details of this, but you should be able to see how it connects various parts of an HTML page to detailed styling instructions.

```
body {
 background-color: white;
 color: #333333;
}
body, p, ol, ul, td {
 font-family: verdana, arial, helvetica, sans-serif;
 font-size: 13px;
 line-height: 18px;
}
h1, h2 {
 border-bottom: dotted purple;
}
h1 {
 font-family: monospace;
 font-size: 30px;
 color: red;
}
```

- The first block states that everything in the `<body>` (the main part) of the page should be dark grey on a white background. Note that this time some colours are given as names (e.g. `white`) and one in hex (`#333333`).
- The second block defines the font to use for various parts of the page: paragraphs, lists, etc.

- The third block says that first- and second-level headings should be underlined in dotted purple.
- The final block says that `<h1>` elements should be displayed in large, red, monospaced type.

Figure 5.22 is what you get if you apply the style to the early draft of my home page in Section 5.2.



Chris Bissell's Home page

I've been teaching and researching at the OU for quite some time, in fact **I'm well into my fourth decade!**

Professional Interests

I have contributed to many OU modules over the years, most recently TM355 *Digital Communications*.

My research interests are in engineering modelling, engineering education and the history of technology.

Leisure Interests

I am interested in all types of music, particularly classical, jazz, blues and some country, and I play the piano. I'm also a fan of art, and like to visit galleries and exhibitions.

Figure 5.22 Screenshot of home page with CSS applied

This is both an *example* of a stylesheet and a *warning*. The lesson is, think very carefully about your choice of style: colours, font, character size, underlining, and so on. I think few people would consider this an attractive web page 'look'! (You'll learn more about web page design in Block 1 Part 6.)

5.5.1 Hexadecimal

As you have seen, web page colours in the examples of this section can be specified by a character string of # followed by six hex digits.

Hexadecimal notation takes all possible 4-bit binary words and codes them using sixteen symbols: the familiar denary digits 0–9 plus the letters A–F. (You can normally use either upper- or lower-case letters in a computer program.) There is some further information on hexadecimal notation in the *Using numbers* booklet.

Binary	Denary	Hexadecimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

For larger numbers, you simply use longer hexadecimal forms. Each group of four binary digits is converted to one hexadecimal digit.

Binary	Denary	Hexadecimal
0000 1111	15	0F
0001 0000	16	10
0001 0001	17	11
0001 0010	18	12
…		
0001 1111	31	1F

and so on.

Activity 5.12 (self-assessment)

(a) Continue the first part of the table above from 19 to 24 denary. (Note that hex numbers are often indicated by the subscript 16 – for example $1F_{16}$.)

(b) What is hex FF as a binary code and the denary equivalent?

Activity 5.13 (exploratory)

Experiment with changing the colour codes in the HTML example you met above:

```
<style>
body { background-color: #ffbf80;}
h1 { color: #006600;}
p { color: #6600cc;}
</style>
```

5.5.2 Conclusion

This section has introduced stylesheets, and demonstrated some simple examples. You have been introduced to the hexadecimal system, and how it is used to code a huge range of colours on a web page. The section is designed simply to give you experience in examining a stylesheet and identifying some of its characteristics; you are not expected to design stylesheets for yourself.

5.6 XML

Finally, in this section I want to introduce very briefly what is known as **XML (extensible markup language)**. XML is essentially a way in which information can be marked up and structured to reflect its meaning, very like the examples of data structuring in Part 4. 'Extensible' means that XML is a markup language that can be extended by its users. Using XML, people and organisations can define their own tags so that they can exchange standardised information with other people and organisations. As this is the purpose of XML – to describe the information content of a document in an agreed way – the most important ideas are those you have already studied in Part 4. As you read this section, try to reflect on the material of Part 4, and make a few notes on the relationship between those ideas and the principles of XML.

5.6.1 Using XML

In XML each piece of data is marked with **tags** that identify it. I'll use the example of contacts in an address book to explain this.

Table 5.1

Bill Smith	Jill Jones
123 High Street	100 Main Road
Aberdeen	Truro
AB12 3CD	TR10 2ST

Table 5.1 shows the address book entries for two example contacts Bill Smith and Jill Jones. Each entry has a name and an address. The name has two parts, which I'll describe as 'given name' (e.g. Bill or Jill) and 'family name' (e.g. Smith or Jones). The address also has several parts: house number; street; town; and postcode.

Using XML, the contact information for Bill Smith might be marked up as follows. In this example, I have used tags which I have named myself. Hopefully the names I have chosen will seem helpful.

```
<contact>
  <name>
    <givenName> Bill </givenName>
    <familyName> Smith </familyName>
  </name>
  <address>
    <houseNumber> 123 </houseNumber>
    <street> High Street </street>
    <town> Aberdeen </town>
    <postcode> AB12 3CD </postcode>
  </address>
</contact>
```

Note: In computing, a common naming convention is to use an upper-case letter where there would be a space between words; typically the first letter in the name is lower case. This is the convention I have used.

The entire set of information for the address book entry is enclosed between opening and closing tags which I have named `contact`. The information is then broken down and tagged further to identify parts of the name and then parts of the address.

Activity 5.14 (self-assessment)

(a) In the XML extract above, which two pairs of tags are used to mark up data about the contact's name?

(b) Which pairs of tags are used to mark up data about the contact's address?

To help your understanding I will now go through the XML code in a bit more detail.

- The `<contact>` element describes an address book contact. All of the contact details are placed between the `<contact>` and `</contact>` tags.
- The `<contact>` element contains two further elements: `<name>` and `<address>`.
- The `<name>` element then contains two other elements: `<givenName>` and `<familyName>`.
- Similarly, the `<address>` element contains four further elements: `<houseNumber>`, `<street>`, `<town>` and `<postcode>`.

This hierarchical structure, which is typical of XML, is displayed in Figure 5.23. In this diagram, you can see the relationships between the elements (shown in the rectangles). The actual data values are shown in the ovals at the bottom. Hierarchical structures are often described as 'tree' structures, though the tree is upside down (in the sense that the root is shown at the top and the branches below; family trees are usually displayed in this way).

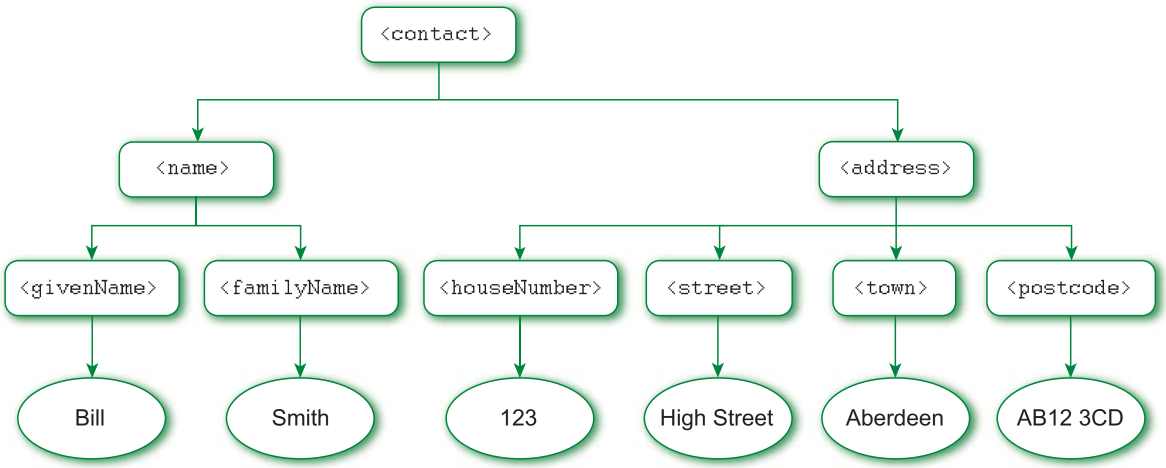


Figure 5.23 Tree structure for the XML elements for address book contacts

Activity 5.15 (self-assessment)

Write out a similar XML extract for the second address book contact: Jill Jones (see Table 5.1).

Clearly, this has only scratched the surface of XML. The only assessment of it in TM111 is indicated by the above activities. If you want or need to know more, there are many tutorials available on the web.

5.6.2 XML vocabularies

The introduction of XML means that people and organisations can get together to develop an XML vocabulary. This allows people with common interests to describe the types of information they wish to exchange in a consistent way. For example, an XML document that describes books can be exchanged between a publishing house, a printing company and a bookseller, provided they are using the same XML vocabulary.

Many XML vocabularies have been defined, each dedicated to describing a particular form of information. Two examples are given below.

- *Mathematical Markup Language* (MathML) permits mathematicians to exchange and process mathematical data.
- *Open Financial Exchange* (OFE or OFX) is a vocabulary that enables a range of financial programs to exchange information.

5.6.3 Conclusion

This section has discussed the use of XML to provide context and meaning to web page elements. XML is particularly important for communication between computers, and to this end specialist XML vocabularies can be used.

Summary

Hopefully this part has removed some of the mystery of how the web works.

I began by introducing HTML, which is the language of tags used to describe information on the web. You created quite a detailed web page and saw how the different tags affect what you see. I then discussed the idea of hypertext and hyperlinks, and you created some hypertext documents.

I also looked at some newer features of HTML5, which can be used to embed media elements such as audio and video into web pages. I showed how stylesheets can change the appearance of a web page, and you had a chance to try this for yourself.

Finally I presented some of the main characteristics of XML.

This part has covered:

- how a browser accesses a given website
- the structure of a URL
- basics of HTML code, including tagging, hyperlinks, embedded audio and video, and iframes
- using a text editor to write HTML
- Google Sites and the editing tools provided
- the importance of a clear website design that is fit for purpose
- stylesheets and how to change the appearance of a web page
- hexadecimal code words, particularly for defining website colours
- some features of HTML5
- some of the main features of XML and how it relates to data structures.

Answers to self-assessment activities

Activity 5.3

(a) HTML stands for *hypertext markup language*.

(b) *Device-independent* means that someone using HTML does not need to be concerned with what kind of device will be used to display a document. As devices vary widely, this allows the browser to adapt a display to the type of device being used, leaving the author free to indicate general styles for text.

Activity 5.6

(a) Nesting is the practice of enclosing a pair of tags within another pair of tags. You can nest HTML tags, but you must remember to close the inner tags before closing the outer tags.

(b) No. Some tags are useful but optional, and most browsers can tolerate their absence and still work properly.

(c) Yes; if they are not, they are ignored by the browser. However, browsers treat upper case and lower case as equivalent.

Activity 5.8

You should recognise the simple tags and their functions: `<h2>`, `<h3>`, `<a>`, `<p>`, `<ul>`, and `<li>`.

There are also a number of tags probably new to you such as: `<head>`, `<body>`, `<div>`, `<div class>` (in particular, `<div class = "outer" / "header" / "nav" / "article" / "footer">`) – although you may have noticed some of these in the HTML generated by Google Sites or if you've looked at the source code of other web pages.

Activity 5.11

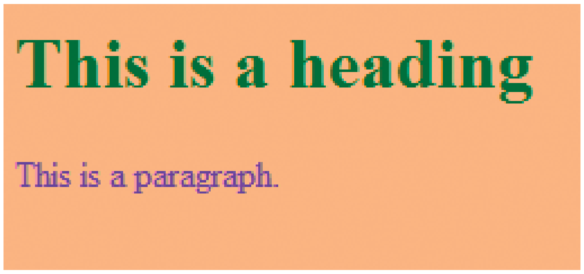


Figure 5.24

The website http://www.w3schools.com/colors/colors_picker.asp gives hex codes for colours. You do not need to know how colours are defined in HTML but if you are interested see the website, which gives a huge number of hex codes. Figure 5.25 shows part of the purple range.

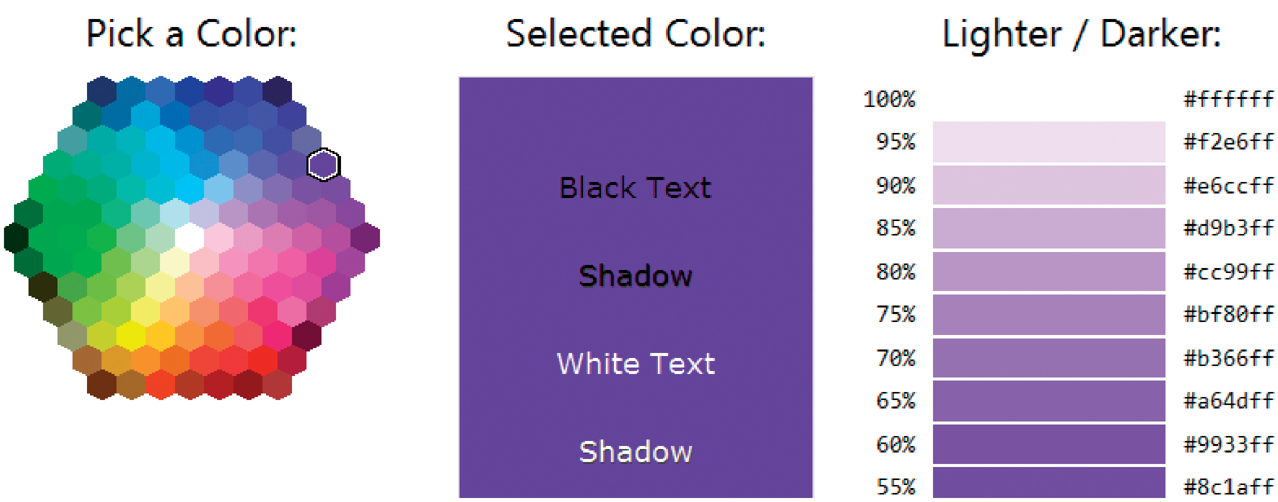


Figure 5.25 Colours defined by hex codes

Activity 5.12

(a) 0001 0011 = 19 = 13_{16}

0001 0100 = 20 = 14_{16}

0001 0101 = 21 = 15_{16}

0001 0110 = 22 = 16_{16}

0001 0111 = 23 = 17_{16}

0001 1000 = 24 = 18_{16}

(b) FF_{16} is equivalent to 1111 1111 which, as an 8-bit binary word, represents $2^8 - 1 = 255$ denary.

Activity 5.14

(a) <givenName> and </givenName>

<familyName> and </familyName>

(b) <houseNumber> and </houseNumber>

<street> and </street>

<town> and </town>

<postcode> and </postcode>

You may also have included:

<name> and </name>

<address> and </address>.

These tags are a bit different because each pair only contains other tags, rather than the data directly, but this issue will not be covered further here.

Activity 5.15

```
<contact>
  <name>
    <givenName> Jill </givenName>
    <familyName> Jones </familyName>
  </name>
  <address>
    <houseNumber> 100 </houseNumber>
    <street> Main Road </street>
    <town> Truro </town>
    <postcode> TR10 2ST </postcode>
  </address>
</contact>
```

Acknowledgements

Grateful acknowledgement is made to the following sources.

Figure 5.3. Taken from https://www.w3.org/TR/html4/

Figure 5.7. Taken from http://www.bedfordshire-news.co.uk/100-years-bedford-luton-archives-records-service/story-21729685-detail/story.html#3

Figure 5.17. © The Courtauld Gallery

Figure 5.5, 5.8, 5.14, 5.16 and 5.23. © Courtesy Chris Bissell

Figure 5.25. Taken from http://www.w3schools.com/colors/colors_picker.asp

Every effort has been made to contact copyright holders. If any have been inadvertently overlooked, the publishers will be pleased to make the necessary arrangements at the first opportunity.

Part 6 Crossing boundaries

by Elaine Thomas and Ingi Helgason

Contents

Introduction

As you have seen in earlier parts of this block, computer systems pervade our everyday lives. We use these systems directly when we are working on our personal computers. However, increasingly we are making use of embedded computer systems where the technology is invisible to us. For example, computers are embedded in home entertainment and gaming systems, mobile phones, MP3 players, digital cameras, domestic appliances such as washing machines and microwave ovens, car ignition systems, electronic turnstiles, and so on.

Given the range of people using computer-based technologies, most of whom are unfamiliar with what goes on inside the computer, and given the wide range of uses to which these technologies are increasingly being put, there is a need to find new ways to interact with computers. An **interface** can take many forms, and a good way to think about an interface is as a 'shared boundary' between the computer and the **user**.

In this part, you will examine the different ways in which humans can interact with computers through the interface. Most importantly, you will learn some of the basic skills and techniques associated with building good interfaces.

Activity 6.1 (exploratory)

There are many types of interface with different purposes and functions. Think about two or three of the interfaces that you encounter in your daily life and for each one consider:

(a) what technology is involved, for example: a screen display, text input, dials, LED or touch-screen

(b) the 'human' aspect, for example: easy, difficult, enjoyable, open-ended, restricted choices, frustrating, complex, exploratory, familiar, unconventional, quick.

Comment

I thought of the following examples. Of course, yours might be very different.

Washing machine:

(a) Buttons (on/off), dials (multiple options), LED display (shows time in numbers).

(b) To me, this is a fairly simple interface with two or three buttons to press and one option on the dial to select. It is simple to me because I use a limited range of options frequently. It was not so simple the first time I used it, and I did have to read the manual.

TV remote control:

(a) Lots of different buttons on one device. Some remotes have pointer controls that can be directed onto the screen itself. Visual feedback on the TV screen, and possibly audio feedback.

(b) This can be a complicated process as the TV has multiple states and menus, for example: 'live' TV, recordings, listings guides, and external screens such as BBC iPlayer. The menu systems are not consistent across these different interfaces, so it can be difficult to remember options that are not frequently used.

My smartphone or mobile phone:

(a) A touch-screen interface, microphone, speaker, voice control.

(b) This is well designed and enjoyable to use. The touch-screen keyboard is a little small. I have disabled the predictive text facility because I sent too many strange messages when it amended my typing without my noticing. I haven't felt the need to use the voice control facility.

Study note 6.1 Referencing

You will notice that I have used and cited the work of different authors in preparing the study material for this part. I have used their work because they have made important contributions to the study of human–computer interaction (HCI).

For each citation in this part, you will find the full reference in the References section at the end of the part. The style for references that is used throughout the TM111 module texts closely mirrors the style you are expected to use for TMAs.

There are, however, some minor differences. Because this is a study text, I have mentioned the title of a book because it is a classic work, but on other occasions I have simply cited an author, for example (Grudin, 1993).

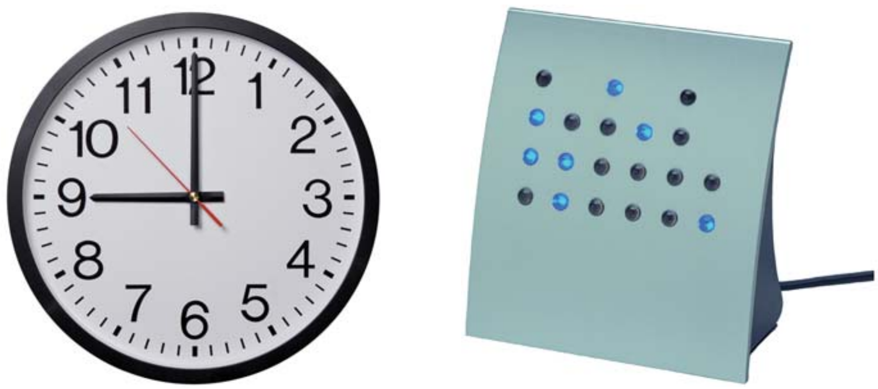
When formatting a reference for a TMA, it is best to consult the OU Harvard guide to citing references, located in the Library's 'Referencing and plagiarism' section.

Computers for us all

6.1

Earlier in this block, you learned that a computer works by converting information into data – more precisely, a series of 1s and 0s – and manipulating these representations in complex ways before converting them back into information in a form that is accessible to the world outside. Without the ability to input information to, and output information from, the computer, it would be impossible to know what operations, if any, the computer had performed. In effect, in the absence of inputs and outputs, a computer is little more than a fancy room heater.

Consider the two types of clock shown in Figure 6.1. Although both types of device can be used to display the time, the clock shown in part (a) is much easier to read.



(a) (b)

Figure 6.1 Contrasting clocks: (a) an analogue clock and (b) a binary clock

Making the output – and input – of a computer-based device easy for humans to understand is important because:

- More people will make use of the device in order to do things. In this sense, the device becomes useful to a larger number and wider range of people.
- More people can use the device, because they understand how it works at the level that they want to use it. In this sense, the device becomes more usable – that is, easier to use – by such people.
- It improves the reliability of the device, in the sense that it consistently behaves in the way that people who use it expect it to behave.

The last point is particularly important. While in principle it is possible to interact with a computer through binary forms of input and output, if we do then it is not long before errors start to creep into the interaction – errors that arise from difficulties with reading, interpreting and writing representations.

Activity 6.2 (exploratory)

Which of the following ways of representing the output from a calculator do you think is the most readable (in the sense of easiest to understand) and most reliable (in the sense of least likely to lead to a mistake)?

1202 (decimal)

MCCII (Roman)

100 1011 0010 (binary)

Comment

Although each of the above represents the same number – i.e. one thousand, two hundred and two – the decimal format is by far the easiest to read and hence is the most reliable of the three representations.

It is important to appreciate that this is because the Hindu–Arabic numerals on which our decimal representation is based are the standard way of representing numbers in the contemporary world. This was not always the case. For example, if you had asked an average sixth-century citizen of the Roman Empire the same question, you would have got a very different answer.

In short, and as you will see later in this part, readability and ease of understanding vary with context – that is, time and place – and with people, their languages and cultures.

The discipline of **human–computer interaction (HCI)**, which you will look at in more detail in the next section, arose from the attempt to shift from the machine-centred – or **technocentric** – form of representation (binary) to a more **human-centred** form of representation in order to eliminate, or at least minimise, the potential for errors of interpretation and improve the reliability of computer-based devices.

As you saw with the clocks example, one important consequence of this is that making something closer to the familiar human conception makes it more accessible to a larger number and wider range of people. In this way, computers not only become more usable; they also become more useful. We might summarise this by saying that HCI is concerned with developing technologies to empower people.

Finding the right form of input or output for a computer-based device is crucial, since it can mean the difference between a successful technology used by a wide variety of people and a technology that is either used only by a select few or, worse, a failed technology that is not used by anyone.

6.1.1 Failures and disasters

Large numbers of software and hardware projects are cancelled before they are completed. Large public and governmental organisations are often criticised in the media for ordering and eventually cancelling expensive computer projects, but the same happens in private industry. Projects are cancelled for many reasons, including running over budget, being beaten to market by a rival, changes in government policy or a sudden change in the market. However, there are also technologies that make it to market, only to end up as spectacular commercial failures.

Activity 6.3 (exploratory)

List several reasons why you think a new computer-based product (either hardware or software) that is introduced onto the market might end up a commercial failure.

Comment

There are a number of possible reasons, but these are the ones that I thought of:

- The product was badly manufactured.
- The product was too expensive.
- The product was not compatible with other products that people wanted to use it with.
- The product did not meet the expectations of the people who wanted to use it in terms of functionality.
- The product was considered simply too difficult to use.

6.1.2 QWERTY forever

In the 1870s, inventors struggled to build a mechanical typewriter. One problem they faced was that the mechanical arm associated with the next typed letter would catch on the previous arm going back. The QWERTY keyboard layout developed in 1874 (see Figure 6.2(a)) was deliberately designed to position arms associated with letters that commonly followed each other as far away as possible from each other. This system worked well; very soon, typewriters were being built based on it, with the first commercially successful typewriter, the Remington No. 2, being introduced in 1878. An industry grew up to educate people on how to use this inefficient yet mechanically effective layout.

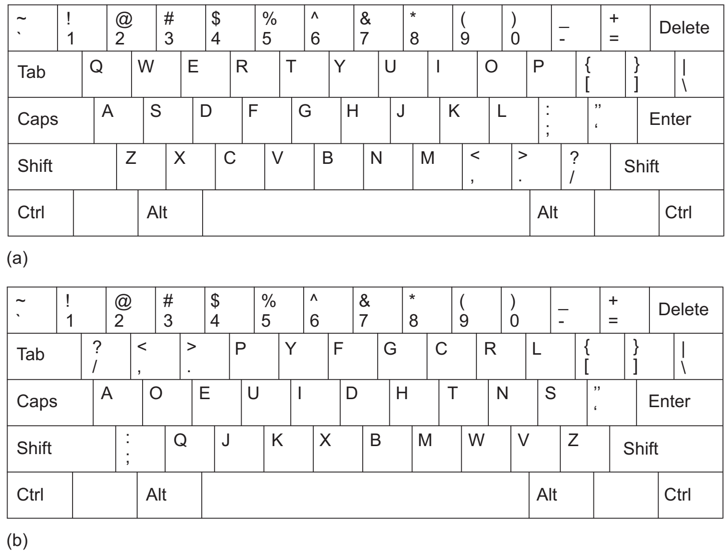


Figure 6.2 (a) The QWERTY keyboard layout and (b) the Dvorak keyboard layout

The world's fastest typist, Barbara Blackburn of Salem, Oregon, could type on average at a speed of 150 wpm (words per minute) using an alternative keyboard layout patented by August Dvorak in 1936 (see Figure 6.2(b)). Her top speed reached 212 words per minute for short bursts. In the 1960s, a new generation of 'golf ball' typewriters was introduced that did not involve the mechanical arm movement that had caused the earlier problems. It was now possible to abandon the QWERTY layout in favour of one that was more efficient for the user. However, this did not happen.

Activity 6.4 (self-assessment)

Why do you think the QWERTY layout was not abandoned, even though the mechanical problems that motivated its adoption were no longer present?

By the time the first digital keyboards were introduced, use of the QWERTY layout had become so entrenched that it seemed only reasonable that these keyboards too should use it. Even touch-screen devices such as the Apple iPhone, where the focus is on gesture-based interaction, include virtual QWERTY keyboards for simple text entry. However, critics have argued that most virtual keyboards are useless for typing, since you can use only one key at a time. Such criticism has prompted developers of the BlackBerry Priv smartphone to include a retractable physical keyboard in their design.

This smartphone appears in Figure 6.3, along with various devices that have used the QWERTY keyboard layout since it was invented. In short, QWERTY lives on.

Figure 6.3 Persistence of the QWERTY keyboard layout: (a) the 1878 patent drawing; (b) an 1870s typewriter; (c) a 1960s teletype; (d) a 1970s terminal; (e) a 1980s PC; (f) a 1990s laptop; (g) a 2000s notebook; (h) a personal digital assistant; and (i) a smartphone with retractable physical keyboard.

This is just one of many examples that show how even a poor interface can, if introduced sufficiently early on, become embedded in the culture using it so that it becomes almost impossible to dislodge. As we develop the interfaces of the computer-based technologies of this millennium, it is important to ensure that those elements that might become embedded in our culture are at their most usable before they are released into it.

6.1.3 Conclusion

Two important principles to bear in mind are:

1. match the technology to the person, not the person to the technology
2. get the interface of a technology right as early as possible, because it has a tendency to live on.

Face to face

6.2

In this section, you will learn more about the interface that must be given to computer-based technologies – or, as it is often referred to, the **user interface (UI)**. However, before I look at the interface, I will first examine two important and related concepts: human–computer interaction and the user. I will then introduce the notion of separability and the idea that different people have a different **model** or *understanding* of how the computer system works. Interfaces can be used to connect two different models of technology – the user's model, which is goal- and task-focused, and the builder or programmer's model, which is focused on the underlying computer system.

6.2.1 Human–computer interaction

In Section 6.1, I made repeated reference to the interaction of people and computers. But what is interaction, exactly?

It turns out that the answer to this question largely depends on who you ask. For designers, interaction is not limited to the connection between humans and computers; rather, it is a way of framing the relationship between people and the objects that have been designed for them, including books, cutlery, cars, washing machines, houses and mobile phones.

For human–computer interaction (HCI) practitioners, interaction is generally conceived in terms of the **feedback loop** arising from the flow of information between a person and a computer system. This feedback loop, shown in the desktop example in Figure 6.4, can be summarised by the example of moving a mouse pointer across the screen towards an icon or menu item. If you were to be asked what happens when you move the pointer to an icon, you might reasonably answer, 'I move the mouse and the pointer goes towards the icon'. In fact, this simple action can be broken down into many steps.

1 The user moves the mouse pointer.
2 The machine displays the pointer's new position.
3 The user compares the new position with the distance and direction of the goal and alters the movement of the mouse to follow this.
4 The machine displays the pointer's new position ...
5 ... and so on until the pointer is over the icon.

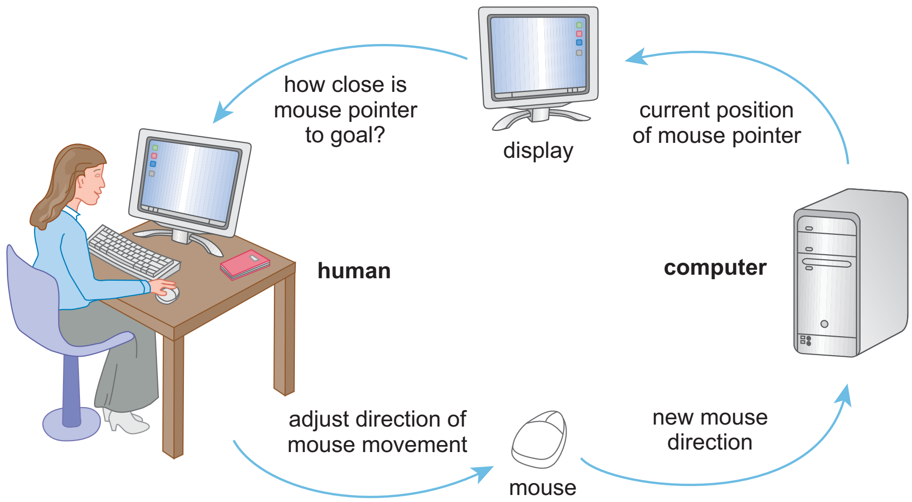


Figure 6.4 Feedback loop between human and computer display

In other words, the user has a goal and acts to achieve it by providing input to the computing system. The user then measures the effect of their action by interpreting the output from the system, comparing the result with the original goal. This cycle happens many times a second – it appears to be a continuous flow – so our simple explanation of 'I move the mouse and the pointer goes towards the icon' appears to be correct.

In addition to being the term used to describe the flow of information between humans and computers, HCI also refers to the discipline that studies how humans interact with computers, their applications and services. Although HCI is a broad subject, it is primarily concerned with the design, evaluation and implementation of interactive computer systems that are safe, efficient, easy and enjoyable to use, as well as functional. HCI looks at the design of computer systems for nuclear reactors, chemical plants, websites, mobile phones, computer games – in fact, all applications and services in which humans and computers need to work together. Various fields contribute to HCI, including computer science, artificial intelligence, philosophy, linguistics, psychology, sociology, anthropology, **ergonomics**, engineering and design.

6.2.2 Users

The term 'user', which has its origin in late 1970s engineering, refers to a person interacting with a computer system in order to carry out a task in pursuit of a goal. It is common in HCI literature and the popular computing press to come across the term 'the user' in connection with discussion of personal and mobile computer-based devices. However, there are problems with this term.

Activity 6.5 (exploratory)

Can you think of a problem with the term 'the user'?

Comment

The term 'the user' implies that there is a single or typical user or range of users. In fact, there might be many different users, each of them unique with respect to the range and type of tasks they want or need to accomplish.

For this reason, HCI expert Jonathan Grudin (1993, p. 114) advises that, whenever possible, we should try not to speak of 'the user'; instead we should speak in terms of 'a user' or 'users'.

Despite these concerns, I will continue to refer to 'the user' since it is a term that has become widespread.

6.2.3 User interfaces (UIs)

The user interface (UI) is simply that part of a computer system which enables interaction between the person and the computer. (It is also known as the human–machine interface (HMI).) However, more often than not – unless you are an engineer, for instance – the two-way communication that takes place during interaction through the UI is all that you will see and understand about the computer system with which you are interacting. As users of computer-based technologies, we are generally only concerned with the UI rather than with the underlying computing system.

For example, ideally, when interacting with your mobile phone to dial a number, you should not have to concern yourself with the underlying communications hardware and software (and other things like antenna patterns) that enable you to make a call from your phone to another. Your concern is with how easy it is to dial the number and establish the call by interacting with the UI of the phone. As long as the two-way communication works and allows you to complete your tasks, you should not need – or *want* – to know what is happening beneath the UI, or how it happens.

We might summarise this by saying that to us, as users, the UI often *is* the system because a user's view of a computer-based device is often limited to, and based solely upon, their experience with the UI. In short, the UI is where we get face to face with the computer, as shown in Figure 6.5.

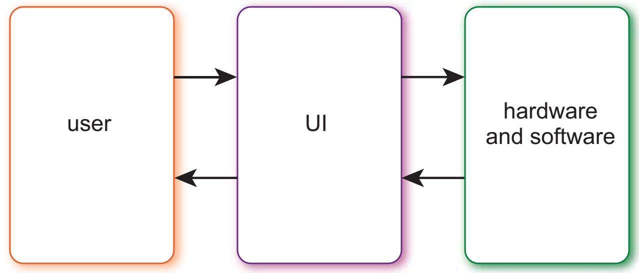


Figure 6.5 The user interface (UI)

The UI is that part of a computer system that a user interacts with directly in order to use a computer-based device to carry out certain tasks. In order for this to be possible, the UI must provide a means of

- *input*, enabling a user to manipulate the system
- *output*, enabling the system to indicate the effects of a user's manipulation.

Consider a microwave oven. It has a panel with a set of buttons that a person can use to adjust the cooking time and heating level, and operate all other functions that the oven provides. These are the means by which to input information *to* the system. Microwave ovens also have a display in order to communicate information to the user – how much time remains, current heating level and so on – and this is the means by which to output information *from* the system. The panel of buttons and the display together constitute the UI of the microwave oven.

Activity 6.6 (exploratory)

Try to identify the UI of (a) your PC or laptop and (b) your mobile phone. Briefly describe two tasks that you carry out by interacting with the UI of your mobile phone or smartphone.

Comment

The UI of my laptop comprises the following:

- on–off switch
- screen display and its contents
- keyboard
- trackpad
- webcam
- speakers
- microphone
- two USB ports, one of which is being used to connect a printer/scanner/copier.

The UI of my smartphone comprises the following:

- on–off switch
- touch-screen display with on-screen keypad

- microphone
- speaker
- camera
- display
- speech interaction, for example, Siri or Cortana.

You might have noticed that in all the previous examples – microwave oven, PC (or laptop) and smartphone – the interaction between users and the system involves two-way communication and feedback: a user interacts with the system by entering some information and the system responds by updating its display. All three examples involve the feedback loop described in Section 6.2.1, with information flowing through the UI from user to system and from system to user.

You might also have noticed that the UIs of different computer systems are different because the ways in which we interact with them are different. In turn, the forms of interaction are different because the tasks we carry out using different computer-based devices vary. For example, a digital watch will not have buttons for setting the heating levels since that is not part of timekeeping; PCs have a display screen, a mouse and a keyboard, reflecting the need to be able to edit documents, produce graphics and so on. With the proliferation of 'smart' personal devices that are always on, it is often the device that alerts the user to do something (take medication, walk more steps, answer a call) through bleeps and buzzes. The user might set the overall goal, but the computer is in charge of managing the tasks and controlling the interaction.

Since the tasks determine the two-way communication that needs to take place, it is the tasks that determine, more or less directly, what the UI for a computer system should be. Therefore, a UI needs to be appropriate for the range of tasks it is being used to accomplish. However, since a computer system may have to serve the needs of a diverse range of users, it may need several different UIs, each of which is right for a particular user. In short, one size does not fit all.

Like 'user', the term 'user interface' has its origin in late 1970s engineering and its meaning has evolved over time. There are a number of problems with this term (Grudin, 1993):

- UI is usually taken to refer to a computer's interface with the user, rather than a user's interface with the computer.
- The UI is technocentric, rather than human-centred: the computer is assumed, whereas 'the user' has to be specified.
- There is a tendency to identify the UI almost exclusively with input and output devices (such as keyboards, mice and displays) and the software controlling them. We rarely consider the network of human support provided by documentation, training, system administrators and customer

support such as technical helpdesks as part of our interface with the system.

In order to address these problems, Grudin suggests that instead of using the term UI, we adopt a more neutral term such as 'interface' or 'human–computer interface'.

Despite the problems mentioned above, in what follows I will continue to make use of the term user interface (or UI) since it has become widespread; however, it must be remembered that there is no such thing as 'the' UI, and that a UI is better understood as a human–computer interface or, more simply, an interface.

6.2.4 Separability

Consider an instruction booklet for assembling a piece of furniture, such as an IKEA table. At the beginning of the booklet there is a symbolic representation of all the different parts and the tools required to assemble the table correctly. Each part has been manufactured in such a way that it will fit together with only a small number of other parts, and then only when they are assembled in a specific order. However, it is important to appreciate that any two identical parts – for example, two screws or washers – can be interchanged without affecting the assembly of the product. This is possible only because the parts conform to a pattern or template and an agreed set of standards specifying how one figure can connect to – or, *interact* with – another.

The use of patterns and templates in engineering, automation and mass production makes specific parts less relevant. Provided the interface (pattern or template) is correctly specified, any parts meeting that specification will work together. This 'plug-and-play' approach to construction is based on the notion of **separability**, which refers to how an interface separates what lies on either side of it so that changes can be made on one side without affecting the other. If we think about this in terms of different models – that is, different ways of understanding how something is done – we might say that a user's model, which is concerned with tasks, does not have to match the builder or programmer's model, which is concerned with the underlying technology.

One important consequence of separability is that models can change across the interface. For example, when you make a landline phone call, the interface requires that the receiver is lifted and the number is entered on the phone. A digital signal is then transmitted through the telephone network along either metal wire or fibre-optic cables so that one telephone line is physically connected to another. While this interface is similar for a mobile phone, the process involved behind the scenes is much more complicated in that the voice data is transmitted digitally via radio waves. For the person using a landline phone or a mobile phone, the interface may have its own peculiar complexities – for example, where phone numbers should be stored. However, the phone engineer has a completely different view or model of the process.

6.2.5 The interface between models

When I use the term 'model' in this section I mean it in the sense that a person holds an internal, 'mental' model or understanding of how a computer system works. Allowing the models of users to be separated from the models of builders and programmers is both the greatest advantage and the greatest disadvantage afforded by the invention of the user interface. This is because problems can arise when users and builders or programmers have different models of the same system.

In his classic work *The Design of Everyday Things* (1988), HCI and design expert Donald Norman asks his reader to consider the following scenarios:

> If you are in a cold room, in a hurry to get warm, will the room heat more quickly if you turn the thermostat all the way up? Or if you want the oven to reach its working temperature faster, should you turn the temperature dial all the way to maximum, then turn it down once the desired temperature is reached? Or to cool a room most quickly, should you set the air-conditioner thermostat to its lowest temperature setting?
>
> *Norman, 1988, pp. 38–9*

As users, we tend to think of the thermostat in heaters and air conditioners in one of two ways: either as a timer – the higher we set the control, the longer the power flows – or as a valve (like that in a tap) – the more we turn the control, the more power flows. Both of these are user models of the operation of a thermostat, and both of them are wrong.

In actuality, turning any of these devices to their maximum output has no effect on the *rate* at which temperatures will change. Each of these devices contains a thermostat, a simple heat-sensitive switch. When the thermostat is 'on', full power flows to the heater (or chiller), which continues to work at maximum power until the correct temperature is reached – at which point, the thermostat switches off and the heater (or chiller) stops working entirely.

Users carry a large and complex baggage of experience gained from using other devices (often older, mechanical devices), and a diverse set of real-world knowledge that they apply to their interaction with the next device in hand. As illustrated in the case of the thermostat example, this causes problems for the designer attempting to understand and model the device.

Activity 6.7 (self-assessment)

Consider the icon-based UI shown in Figure 6.6, where each icon in the main part of the window represents either a document or a folder. Documents are grouped into folders that open a new window when double-clicked. When you right-click on a document and select the print option from the menu, the document is sent to the printer; if you click on a number of documents, all selected documents are printed.

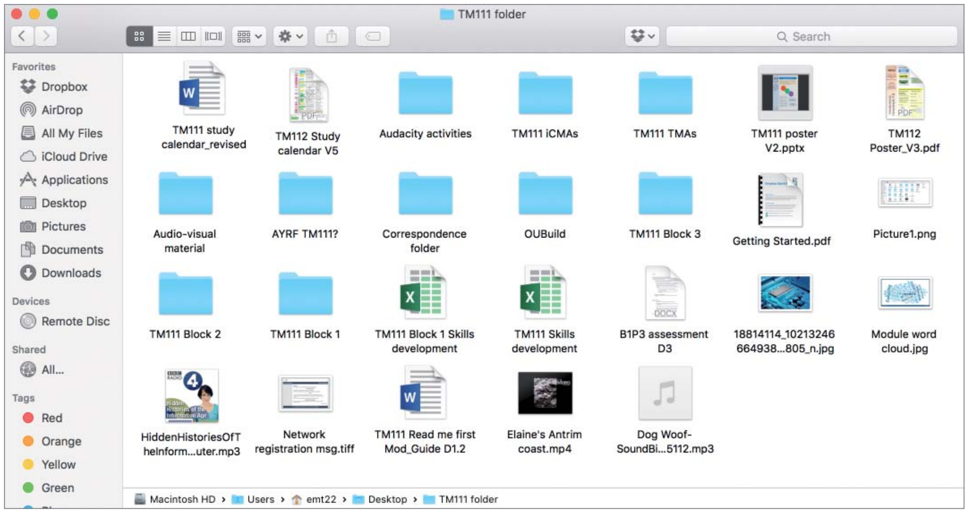


Figure 6.6 Part of an icon-based user interface showing a collection of documents in a folder

If you were now to click on a folder and select the print option from the menu, what would you expect to happen and why? The options are either:

1 all the documents in the folder should be printed, or

2 a list of the documents in the folder should be printed.

Write down your answer before looking at mine.

Before concluding this section, try to answer the following review question.

Activity 6.8 (self-assessment)

Write an explanation of the terms ‘human–computer interaction’, ‘interface’ and ‘separability’. A sentence or two for each term is sufficient.

6.2.6 Conclusion

The important points to remember from this section are:

- there is no such thing as ‘the user’; at the very least, there is ‘a user’ or ‘users’, while at best there are people in particular roles
- users interact with computers through interfaces, and different interfaces may be needed for different users, and different contexts of use
- interfaces allow a user’s model of the task to be separated from the builder or programmer’s model of the underlying system
- the mismatch between the user model and the builder or programmer model can create problems, and it is the responsibility of the interface

designer to attempt to resolve this mismatch through the use of HCI techniques.

6.3 Emergence of the interface

In Part 2 you saw how human–computer interfaces developed over the four generations of computers, from using cards and tape to command-line interfaces to graphical user interfaces (GUIs). In this section I will show how user interfaces have developed since the 1980s and the introduction of GUIs. I'll illustrate these developments as changes in the relationship between the user's model and the builder or programmer's model.

6.3.1 The 1980s: graphical user interfaces (GUIs)

With the command-line interface, which you saw in Part 2, human–computer interaction is textual and language-based – in short, conversational. Having to remember a large number of often cryptic commands placed a heavy burden on the user who had to learn them. For example, in the Unix operating system, the command to list the contents of a directory (that is, show the documents in a folder) is 'ls'. Similarly, 'cp' is used to copy a file and 'mv' is used to move a file.

However, this conversational **metaphor** was retained in the next phase of interface development, with the focus shifting to how one might interact in a foreign country when one did not understand the language spoken. In this context, people might point or gesture at what they wanted, rather than learn and recall the word from memory or have recourse to a phrase book. This led to a shift from conversational interfaces to **graphical user interfaces (GUIs)**.

One of the consequences of the emergence of the GUI was the separation of the programmer's model from the user's model, as shown in Figure 6.7. Users no longer need to know anything about typography or graphic design to use a modern word processor or desktop publishing application; programmers don't have to understand anything about how to write a good novel or an effective office memo. Few people know how an application such as Microsoft Word works in terms of how the software is handled by the computer system, nor do they need to. Programmers are free to modify the software secure in the knowledge that, provided the interface remains the same, the user will be unaware of the change.

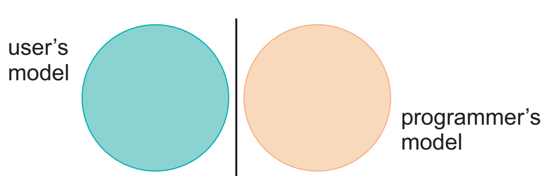


Figure 6.7 Separation of user and programmer models in GUIs

The separation between the programmer's model and the user's model made it possible for computers to assume any form. However, as you saw in Section 6.2.5, it is the difference between the two models that can lead to problems, and this is what makes the process of designing software and interfaces so challenging.

One attempt at addressing the design problem involved the use of metaphor. Using this approach, a GUI calculator should look like a real-world calculator, as shown in Figure 6.8; similarly, dragging a document into a wastebasket icon would delete it, dragging it onto a photocopier icon would duplicate it, while dragging it onto a printer icon would print it.

Figure 6.8 Microsoft Windows 10 calculator application

With the GUI, an example of which is shown in Figure 6.9, the computer started to become a tool for the ordinary person rather than a component of business. One important consequence of this was that software was now being purchased by people for use in their own homes, which meant that software designers had to think about how to create software that people wanted to use, rather than have forced upon them as part of their job.

Figure 6.9 Mac OS X GUI

For example, a company whose business involved writing a lot of letters might previously have had a bespoke piece of letter-writing software built specifically for its use. For the designer, this meant analysing workflow patterns in order to produce an application that was extremely efficient, yet highly specific to the customer's needs. With the rise of the GUI, software changed from specific business-oriented uses to generic non-business ones. This resulted in the appearance of the graphical word processor, which could be used to write not only letters but novels and memos as well. The generic nature of this type of application, along with the need to create desirable software, now forced designers to consider issues beyond the mere efficiency of bespoke business software.

The characteristics of this era of computing are summarised in Table 6.1.

Table 6.1 Characteristics of the GUI era

Interaction technology	Keyboard, mouse, graphical interface
Key terms	GUI
Computers per user	1
UK users	Hundreds of thousands

6.3.2 The 1990s: web user interfaces (WUIs)

The first graphical web browser (shown in Figure 6.10) made its appearance in 1993 and introduced an intermediate model between the user and the programmer. In this model, which is shown in Figure 6.11, the programmer's model consists largely of scripts running on a server and communicating with the user through the intermediate browser, which means that the interface has to be recast in the medium of a document.

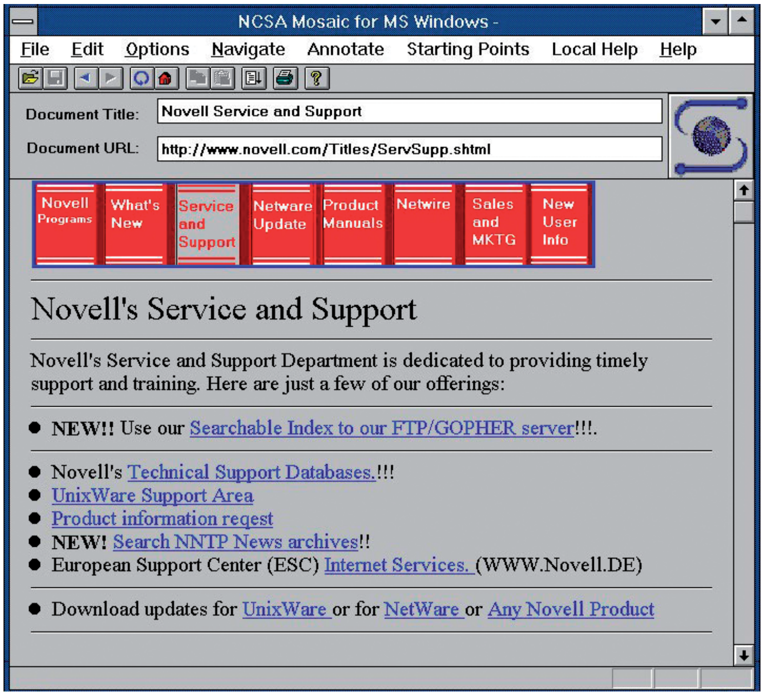


Figure 6.10 Web page with links shown in Mosaic, the first graphical web browser

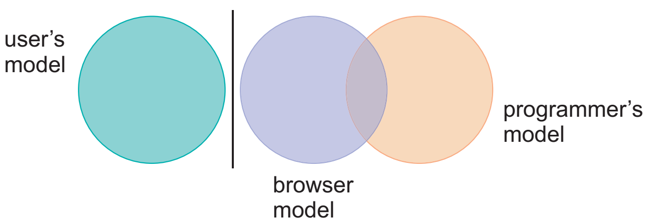


Figure 6.11 Introduction of browser model in WUIs

With the rise of the web, designers were faced with new challenges. While GUIs required the designer to understand the user, **web user interfaces (WUIs)** made any member of the public a potential user, and so it was hard at this stage to understand what prior models they might have in mind.

The characteristics of this era of computing are summarised in Table 6.2. In this era users might have access to several different computers at work and at home.

Table 6.2 Characteristics of the WUI era

Interaction technology	Web browser, mouse, keyboard
Key terms	Human–computer interface
Computers per user	5
UK users	Millions

6.3.3 The 2000s: social and mobile interaction

The rise of the WUI gave people the freedom to indulge in the incredible range of activities we see today. Services such as Facebook, LinkedIn, Flickr and Twitter exist for a variety of reasons, including fulfilling a more playful and engaging role rather than achieving a set goal.

Partly to reflect this shift in the use of computers, HCI came to refer not only to interactions with a computer but also to using a computer to interact with other humans, leading to the interaction situation shown in Figure 6.12.

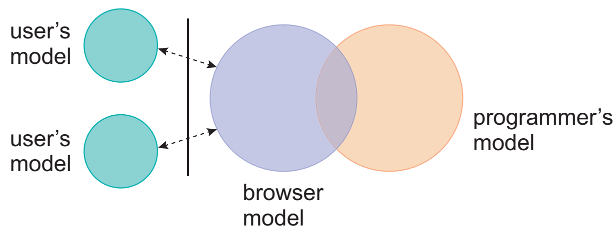


Figure 6.12 Three-way interaction model in the web

Perhaps the most obvious interaction issue is the small screens of portable devices. Conventional applications such as word processors, spreadsheets and even email programs are designed for large displays and attempts to cram them on to smartphones have met with mixed success. Further problems are caused by our limited ability to accurately identify and interact with small objects. As devices have become smaller, the elements of the interface such as buttons and menus have become correspondingly larger.

One way of overcoming these issues is to embed **a tangible interface** (or touchable interface) within, or wrapped around, a computing device such as a smartphone or a tablet so that the physical device itself becomes the interface through sensor technologies embedded in it. A good example of this is the Apple iPad, which has a large, multi-touch, touch-screen display and innovative software, allowing you to control everything using your fingers to make gestures. You can type using the predictive keyboard (Figure 6.13), scroll through photos with a flick, glide through albums, or zoom in and out of a section of a web page. An accelerometer detects when you rotate the iPad from portrait to landscape (and vice versa), then automatically changes the contents of the display to the appropriate orientation.

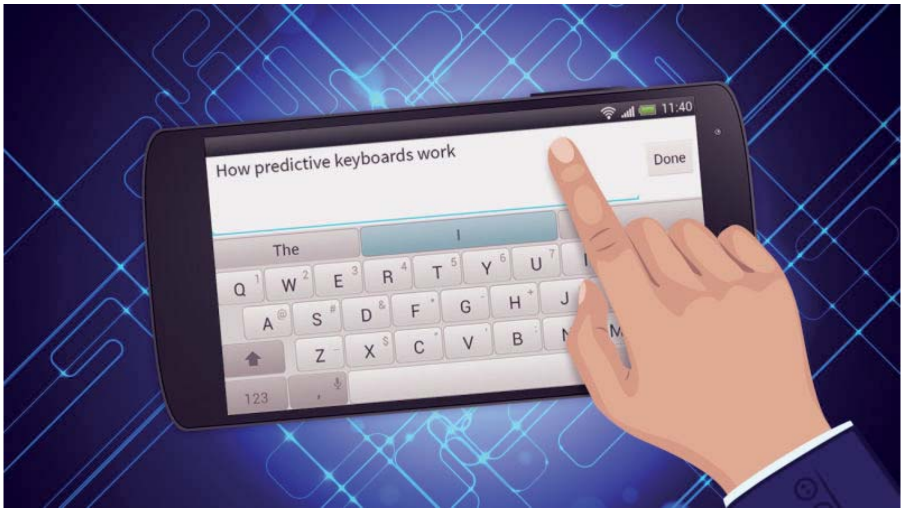


Figure 6.13 Predictive keyboard on the Apple iPad

Gesture-based interfaces using hand movements as input controls have also been used in the context of computer gaming systems, most notably Nintendo's Wii system (Figure 6.14) and Microsoft's Kinect, a controller-free motion-detection system developed for the Xbox 360 platform and released in 2010.

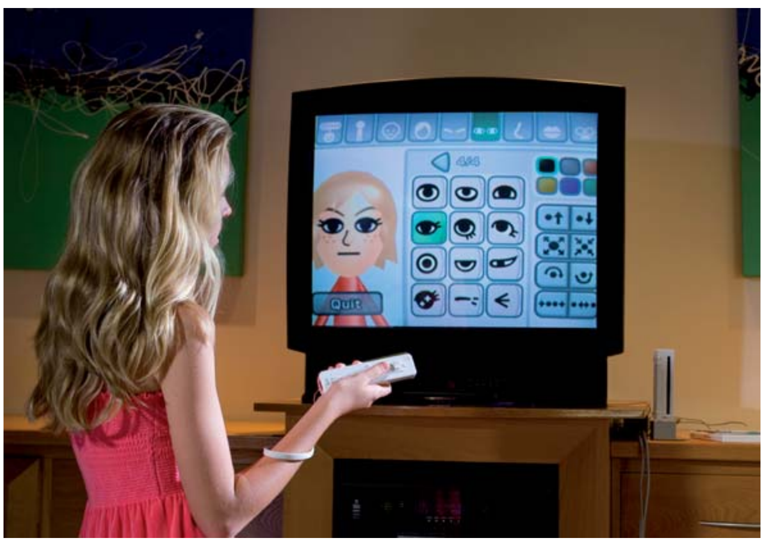


Figure 6.14 Nintendo's Wii system

The characteristics of this era of computing are summarised in Table 6.3.

Table 6.3 Characteristics of the three-way interaction era

Interaction technology	Mobile, web, wireless
Key terms	Human–computer interaction
Computers per user	10
UK users	5–10 million

When computers are embedded in objects or in the environment, they disappear from view. To most users, the interface is the computer; this means that interfaces should also disappear from view. Interaction should no longer be thought of as involving humans and computers, but rather as occurring between human beings and the environment.

Put another way, we might say that in this situation, the interface not only becomes distributed over a number of programmer models (the heating of your house, the film you are watching, the house surveillance system, communicating with members of your family, your children's homework, your office messages, your car's annual safety test or MOT), but also is to be used and shared with a number of applications and people, where each person may have their own model as shown in Figure 6.15. This new kind of interaction – and interface – requires new skills and techniques that are loosely brought together under the idea of **interaction design**, which is the study of how people interact with each other through technology.

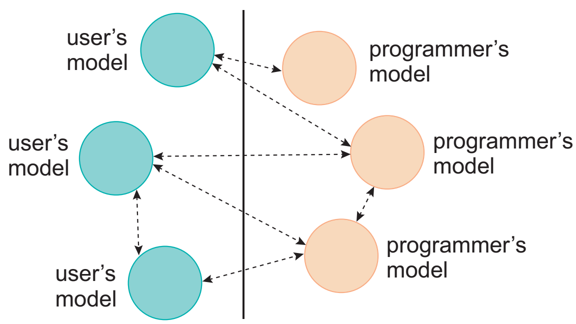


Figure 6.15 Multi-way interaction model in ubiquitous computing

Activity 6.9 (exploratory)

Imagine that you are on a trip away from home in a different country, and you want to have discussions with:

1. a close family member
2. a group of work colleagues.

For each of these people, make a list of the types of software application and the devices that you might choose to contact them. Consider the technology that you have available, and the technology that they might have.

Comment

To call home to a close family member, I might use my smartphone using the in-built voice or video apps, such as FaceTime for iPhone, or other apps like Skype or Viber. Alternatively I might use the messaging and video features that are included in social media such as Facebook, if we both have accounts with these applications. The choice of methods would also depend on the availability of Wi-Fi or 4G. My family member, particularly if they are a young child or quite elderly, might not have an up-to-date smartphone, so I might call to their home landline from my smartphone.

To contact a group of work colleagues for a meeting, I might use a laptop, and use video or voice services, such as Skype or Google Hangouts, or perhaps my company has a video conferencing service that they use, for which I have a subscription or login. These would support group video conferences and might also provide text communication functions too. My colleagues might be using desktop computers with good internet connections, but if I am in a hotel or a café, my connection might not be so good, and it might be necessary to restrict the call to voice only, to save on bandwidth.

You might well be able to think of other apps and services, used on a variety of types of devices. The key point here is that there are lots of different interfaces used to carry out very similar tasks, and more are appearing all the time. To be competitive, an interface has to be appealing and useful to potential users, and it may have to work 'seamlessly' together with other applications and interfaces.

6.3.4 Conclusion

In this section you studied the emergence and evolution of the interface over time. You also learned that the continuing development of the interface has led to a separation of the programmer's model of the system from the user's model of his or her task.

6.4 Design

I will begin by considering what design is and why it is important, and briefly look at examples of good and bad design. You will then learn some design principles and techniques, explore what is meant by **interaction design**, usability and accessibility, and learn how to carry out a **user evaluation**.

6.4.1 Why design is important

Put simply, design as an *activity* refers to the process of originating and developing a plan for a device, product, service, structure, system or component on purpose. One way to become clearer about the meaning of design as a process is to compare it with engineering, because although there is no clear boundary between the two, there is a critical difference in perspective. All engineering and design activities call for the management of trade-offs. In classical engineering disciplines, these can be quantified: material strength, construction costs, rate of wear, and so on. In design disciplines, by contrast, trade-offs are more difficult to identify and to measure because they rest on human needs, desires and values.

One of the basic goals of HCI design is to improve the interactions between humans and computers by making the computers – which means interfaces – more usable and receptive to the needs of users. The design of an interface affects the amount of effort that users must expend in providing input to a system, interpreting output from the system and learning how to do both these things. Presenting data in a way that provides useful information to users allows them to make sense of their interactions with computers and to make appropriate decisions in particular situations. For example, car drivers know what speed they are travelling at by looking at the speedometer on the instrument panel. However, in unfamiliar situations it is not always possible to know what is needed from data in order to ensure that appropriate information can be generated.

Depending on the design of the interface, users of a computer system will determine whether or not the whole system is usable, i.e. easy to learn and use. For this reason, interface design is of critical importance.

6.4.2 The good, the bad and the ugly

Interfaces should be designed to enable users to interact effectively with computer-based devices. We can all think of examples from our everyday lives where we have been confronted with systems having poor interfaces, some of which have given us considerable annoyance. For example, I have difficulty programming my home entertainment system because of the poor design of the remote control. Although this doesn't have disastrous consequences, it does waste my time, resulting in frustration and dissatisfaction with the product. If, however, the data generated by a computing system is not presented at the interface in a form that the user can readily interpret, bad – possibly even ugly – things can happen. For example, poor interface design was identified as one of the causes of the Three Mile Island nuclear accident in 1979.

Bad design can also result in loss of business. (One of the classic cases in this regard is the now defunct online shopping site Boo.com.) For example, if a shopping website is not easy to use and does not support prospective customers in their buying process, then the customers will go elsewhere – either to a rival shopping site or to a high-street ('bricks and mortar') shop – resulting in a loss of business for the site.

To design a good UI for a computer system requires a good understanding of the users, their goals, their tasks and the specific contexts of their use of the technology. Fortunately, there are a number of design principles that we can use to improve the design of user interfaces, making their use more efficient, more effective and more satisfying to their users – in short, making them more *usable*.

6.4.3 Design principles

I will now look at some of the basic principles that are used when designing interfaces.

Affordance

Affordance, in the context of HCI, refers to the attributes of an object that indicate to people how it is to be used. Affordance is concerned with how to use an object, not what the object is for or whether to use it, which are issues of system functionality and context of use, respectively.

Activity 6.10 (exploratory)

What must you do to open each of the doors shown in Figure 6.16?

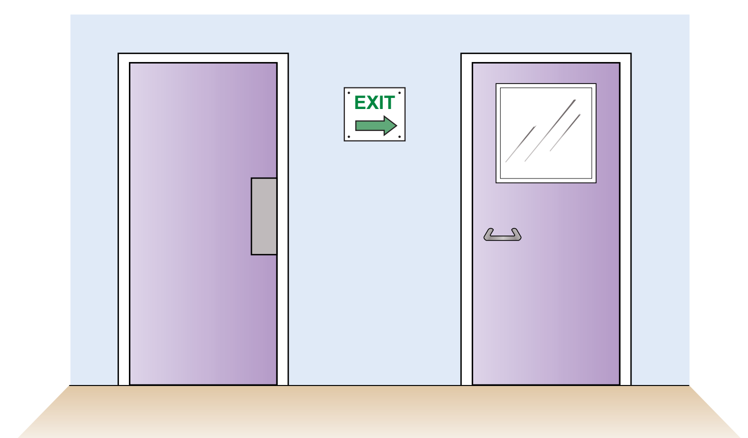


Figure 6.16 The affordances of doors

Comment

In general, plates *afford* pushing and handles *afford* pulling. However, it is important to appreciate that this is not always the case: For example, even in some modern buildings many of the doors have handles, yet must be pushed in order to open them (Figure 6.17). This would be considered by many designers to be an instance of poor design.

(a) (b)

Figure 6.17 (a) Text and handle say ‘Pull’; (b) text says ‘Push’ but handle says ‘Pull’

Activity 6.11 (exploratory)

This activity asks you to consider some affordances in the user interface of an external website. It requires you to be online, so the instructions for it are in the online activities section associated with this part on the module website. If you can’t do it now, please continue studying but make a note to try it when you can.

One interesting – and important – question about affordances is whether they are **intuitive**, by which I mean they seem obvious or natural. Is there a natural way of using an object such that anyone who uses it will interact with it in the same way? Will everyone pull a handle?

This reminds me of a scene from an old Star Trek film in which one of the characters, Scotty, a spaceship engineer who has travelled back in time from the twenty-third century to the present (1986), picks up a mouse and tries to use it to talk to the computer (Figure 6.18). Clearly, in the future, it was considered natural to interact with computers in this way – that is, through a conversational interface.

Figure 6.18 Star Trek's Scotty using a computer mouse as a microphone – another 'natural' way of interacting with a mouse

The lesson to be derived from the Star Trek example is that the affordance of an object is 'intuitive' insofar as the object resembles, or is identical to, something with which the user is familiar. Intuitiveness, at least in this context, has more to do with familiarity than with naturalness. This is important since what we are familiar with is determined by culture and context, both of which must be taken into consideration when designing interfaces for their intended users.

The key points about affordance are given in Box 6.1.

Box 6.1 Affordance

Design principle: All the elements of an interface should have good affordance.

Guidelines: Think about how the element looks and behaves. Just as a door handle affords pulling, a cup handle affords grasping and a mouse button affords pushing, the element's appearance should make it obvious how a user should interact with it. Interface elements should be designed so that it is obvious how they should be used.

Examples: In the Mac OS X GUI shown in Figure 6.9 in Section 3, buttons are given a shadow, suggesting something that stands out from the screen. In this way, pushing the button is the suggested interaction. Another example is the underlining of links on web pages that afford clicking.

When Apple released iOS 7 in 2013, they moved away from the visual affordance of the three-dimensional design of icons, to a style that is described as 'flat design' (shown in Figure 6.19(a)). This trend has influenced many interfaces (see Figure 6.19(b)) and websites, and buttons that look 3D seem quite dated. While this might seem to go against the idea of using a physical metaphor of a real-world button to signify pressing, it may be that we are now so familiar with the icon model of interface design that we no longer need these visual cues, especially when we consider the younger generation of 'digital natives'. Of course, fashion and novelty play a part too, so perhaps we will see 3D design coming back again when it looks nicely retro!

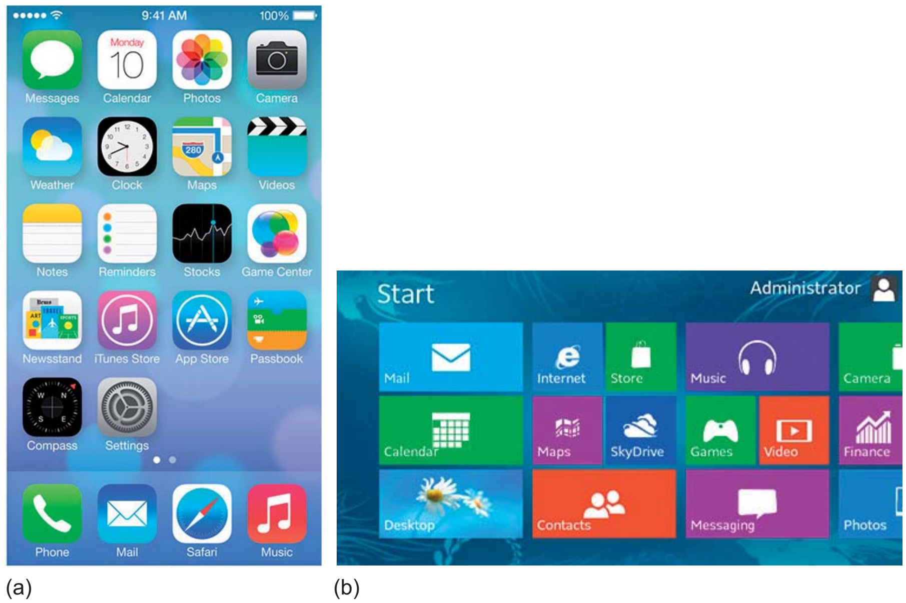


Figure 6.19 (a) Apple's iOS 7; (b) Windows 10 'flat design'

Visibility or perceivability

Activity 6.12 (exploratory)

What do the symbols on the lift buttons shown in Figure 6.20 mean?

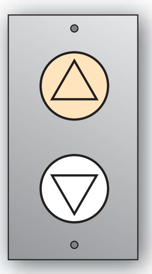


Figure 6.20 Symbols on lift buttons

Comment

The upright triangle indicates travel in an upward direction, while the inverted triangle indicates travel in a downward direction.

Historically this principle is referred to as **visibility**, reflecting that early interaction design dealt mainly with visual interfaces supported on screens.

However, interactive products are no longer just computers on desks with a keyboard, mouse and screen, but take many different forms. In the context of interface design, **perceivability** means making it clear what an interface element is used for. One way to enhance the visibility of an interface element is by labelling it appropriately. For example, the lift buttons shown in Figure 6.20 have been labelled with symbols that indicate the direction of travel: up and down, respectively.

Another way of improving perceivability is through suitable positioning of interface elements. The more visible interface elements are, the more likely it is that a user will know what to do next and thereby move towards achieving their goal. In contrast, functions that are 'out of sight' are more difficult to locate and know how to use, and users will find it more difficult to achieve their goal. Norman (1988) gives the example of a car in which the controls for different operations are visible (for example, indicators, headlights, horn), indicating what can be done. The relationship between the position of the controls and what they do makes it easy for the driver to find the appropriate interface element.

The key points about perceivability are given in Box 6.2.

Box 6.2 Perceivability

Design principle: All the elements of an interface should have good perceivability.

Guidelines: To achieve perceivability, think about the goal that will be achieved by using that element. Are there any icons that could be used to draw the user's attention? Is there a standard symbol for the goal? Is there a single word that describes it?

Examples: In an interface for a software media player, the play button should be visible. Applying the perceivability guideline, there is a standard recognisable symbol for a play button; it looks like an arrow head pointing to the right. (Look at the remote control on a video or media player if you're not sure.) In some email applications, an envelope icon shows that an email has been received.

Feedback

Put simply, **feedback** is some indication that something has happened. For example, when you touch an icon on a smartphone, the display should change accordingly. Feedback should be rapid and continuous; if not, users will repeat actions – for example, clicking a mouse button twice when only once was required – leading to the occurrence of errors. One issue for designers is what type of feedback is appropriate in a given situation: something simple, such as a character displayed on the screen, or something more complex, such as a pattern of blinking LEDs?

Sound is often used as a cue to announce that something has happened so that users are not required constantly to monitor a device or application for changes. For example, the ping or beep made by a microwave oven indicates that it has finished cooking. Here sound is being used to attract attention, perhaps to alert the user that they need to do something; it is also sometimes used to raise warnings, for example, in aircraft cockpits and the control rooms of power stations.

Activity 6.13 (self-assessment)

Give an example of a non-speech sound associated with a computing device with which you are familiar.

Feedback is related to the concept of perceivability. In the context of interface design, it means making it clear what action has been achieved through the use of the interface element. A related design principle is feedforward, which refers to designing in such a way that it is clear what will happen before an action is performed. Feedback is also used to indicate that one part of an action has finished and another can begin.

The key points about feedback are given in Box 6.3.

Box 6.3 Feedback

Design principle: All interface elements should provide adequate feedback in response to a user's actions.

Guidelines: Think about what information should be sent back to the user about their interaction with the interface element. Can you give visual feedback, such as a message or an icon? Is there a single word that says the action has been completed? Is there a standard symbol to give the feedback? Can a simple sound be used to say that something has happened, or tactile feedback such as a vibration, or a combination of these?

Examples: When I set an alarm on my smartphone, a small picture of a bell appears next to the time display, indicating that an alarm has been set. While this is useful feedback, it is not adequate – it would be useful to have the alarm time displayed next to the bell to show the time I had set.

Game controllers often vibrate (tactile feedback) to reinforce a physical feeling of carrying out an action. When I press a key on the touch-screen keypad of my iPhone, there is a pleasant 'click' sound to suggest a traditional physical handset.

You have seen that various kinds of feedback are available for UI design: audio, tactile, visual and verbal, as well as any combination of these.

Deciding which combinations are appropriate for different kinds of activities is an important part of making interaction effective.

Constraints and flexibility

These principles are related to the ability of an interface to prevent errors if possible, or to make errors easy to recover from, if not.

Constraints

This principle is about designing an interface so that it limits the possible actions that can be performed in order to avoid errors or inappropriate actions, either temporarily or permanently. An example would be the greyed-out 'send' button at the bottom of an online form that does not become active until all the necessary information has been filled in.

Constraints can be in the form of physical barriers, such as the greyed-out button that won't react to a click, or a slider that will only move so far and no more. They can also be psychological, using imagery such as symbols or warning sounds, which make us hesitate to carry out an action. These are often based on commonly understood conventions such as red for stop and green for go.

Flexibility

The principle of allowing flexibility in use encompasses the idea of allowing more than one way to carry out a task. It also suggests that the system should include a level of **tolerance** to error by the user. This design principle is particularly relevant for interactive systems that include a level of 'smartness' in their programing, enabling them to adapt to some extent to the user inputs.

For example, a **speech recognition** system for a phone booking system might invite responses to questions. Rather than forcing users to comply with a system operating with a limited range of tolerance – for example, getting people to respond 'properly' by saying only 'yes' or 'no'– a tolerant system would accept a range of possible inputs, including 'okay', 'yeah', 'fine' and 'uh-huh' for 'yes', and 'nope', 'no way' and 'nah', for 'no'.

The key points about constraints and flexibility are shown in Box 6.4.

Box 6.4 Constraints and Flexibility

Design principle: The interface should be designed to reduce the number of errors and facilitate recovery from them.

Guidelines: For an interface to be tolerant of errors, you should think not only of the correct ways that users will perform a task but also of the ways in which users can make errors. Are there choices available to users at a certain point in an interaction that could cause errors? If this is the case, make these wrong choices unavailable. Is there a message that would help a user to confirm a possible wrong choice? Are there examples of good choices that you can provide?

Are there cases where there may be more than one possible correct option? If so, think about how to allow users to provide input in all the ways that are logical, sensible or natural to them.

Examples: To prevent the wrong choice of menu item, some items might be greyed out. Displaying a help message if a user attempts to choose a greyed item can solve this. Some interfaces provide examples to show the type of information that is required. For example, a user could be offered an example of the format in which a date is to be entered.

Other principles

These design principles are not a definitive list. There are many others, but they form a good basic checklist for developing and assessing interfaces. HCI and interface design experts also maintain that it is important to consider the following.

- Grouping: Structure the interface in a way that is meaningful to users. Things that users think of as related should appear together in the interface, or at least they should be clearly and closely associated.
- Consistency: Present the elements of an interface consistently in terms of their appearance, positioning and behaviour. For example, in many applications, you will notice that menu items such as File, Edit, View, Window and Help are the same and function in a similar way.
- Simplicity: Keep things as simple as possible by making simple, common tasks simple to do.

The last point is particularly important. To achieve simplicity, the designer should employ actions and interface controls that are natural for the user and speak in the user's own language. In addition, complex tasks should be broken down into simpler subtasks, and the designer should find out what tasks are most common for users and make these as short and simple as possible for them to achieve.

Keeping it simple is a real challenge for interface designers, as it implies knowing precisely what is natural or familiar. To do it properly and keep things simple means spending time getting to know users and seeing what they find natural. This is sometimes in conflict with the tight schedules associated with interface design.

In addition to these design principles, there are a number of psychological principles that are important for design. These include the following.

- Users see what they expect to see.
- Users have difficulty focusing on more than one activity at a time.
- It is easier to perceive a structured layout.

- It is easier to recognise something from a list of options than to recall it without any hints.

6.4.4 User-centred design (UCD)

In Section 6.3, you learned that there is a separation between the user's model of the task and the programmer's model of the underlying computer system. This separation of models led to a shift in software design. The essence of interface design is what is known as **user-centred design (UCD)**, where users are at the centre of the design model. This might seem rather obvious; however, in many situations the concerns of users are placed at the bottom of a list of priorities. In addition, programmers often make the mistake of thinking that they are users too, and so they design things around what is convenient for them.

Activity 6.14 (exploratory)

During the process of designing a website for a large company, the designer was asked by the company's directors to create categories based upon existing departmental boundaries. On this basis, printers were listed on the hardware page, while printer ink was listed on the consumables page.

Try to identify what is wrong with this approach in terms of user-centred design.

Comment

On a user-centred website, printers and all the information related to them should be located on a single products page, because users are thinking about their own activity of printing rather than about the company's stock organisation practices.

The main idea behind UCD is that users, the people who will be using a product or service, know best what their needs, goals and preferences are. The role of designers is to find out these things and then design with them in mind. Before designing a system, a good interface designer will try to answer the following questions.

- Who are the users?
- What are their experiences?
- What skills do they have?
- What tasks will they be using the system to perform?

Designing an interface might appear to be a highly intuitive activity, but in practice it is not, and it can actually be very structured. Interface design using the UCD approach is based on three principles.

1 An early focus on users and their tasks during the design process (for example, conducting interviews, observing and recording users in action, getting users to complete questionnaires).

2 Empirical measurement (for example, number of errors in using the interface, number of steps taken to achieve a goal).

3 A design process involving iterations of designing, building and evaluating a system so as to incorporate feedback from users, as shown in Figure 6.21.

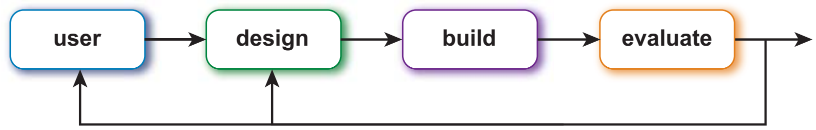


Figure 6.21 The iterative design process

6.4.5 Conclusion

In this section you have studied what design is and why it is important. You've briefly looked at examples of good and bad design. You've also studied some important basic principles that are used when designing interfaces, including:

- affordance
- visibility
- feedback
- tolerance.

You have considered the importance of user-centred-design (UCD), which puts users at the centre of the design process. Finally, you have considered why it is important to use an iterative design process that takes account of users and their feedback at every stage of the design process.

Usability and accessibility

6.5

A good interface is one that is easy to use and easy to understand, meets the needs of the intended users, and supports users in the tasks they wish to undertake. A good interface designer thinks about the users of the interface and pays great attention to the **usability** of the interface for users.

6.5.1 What is usability?

Usability has been defined in a number of different ways, and standards (see Box 6.5) are a way of defining usability.

Box 6.5 Standards

Standards are tightly-defined descriptions of how a product should be constructed or operate in order to be accepted. They provide an agreed specification for doing something, describing how it has been designed, which is published as widely as possible. The modern world is full of standards; you have already met standards for SQL in Part 4 and for HTML in Part 5. Beyond this module, almost every aspect of your life is governed by standards – they are responsible for guaranteeing that machines work as expected, that food, drink and energy supplies meet acceptable guidelines, and that devices can be expected to work with one another. In fact, we rely on standards so much that we rarely think of the consequences when there are none in place.

The International Organization for Standardization (ISO) has overall responsibility for the development of standards for HCI and usability, while Publishing@WC3 is responsible for the development of web standards.

According to Part 11 of ISO 9241, the international standard ‘Ergonomic requirements for office work with visual display terminals’, **usability** refers to:

> The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.
>
> *ISO, 1998*

The above definition refers to the extent of a system’s usability and indicates that it should be possible to distinguish a system with good usability from one with poor usability. Furthermore, if – as stated earlier – the interface is the system, then the usability of a system is the usability of its interface.

The ISO definition raises a number of other important points.

First, for a system to be usable, the interface should be perceived as such by the specified users – that group of users for whom the system has been designed. A system designed to be highly usable by nuclear power plant operators need not be usable by the general public.

What are the goals of the users? What are the users trying to do with the system – does it support what they want to do with it? The interface should support the users in their tasks.

Effectiveness refers to the goals or tasks being achieved accurately and completely.

Efficiency refers to the resources expended to achieve the goal or task, including, for example, time and effort.

Satisfaction refers to the comfort and acceptability of the computer system to its users (and to other people affected by its use).

Can a goal or task be achieved effectively, but not efficiently? Or without being satisfactory? The short answer is yes. Consider, for example, an online ticket booking system in which it takes a long time for a passenger to make a reservation. It is effective in that the reservation can be made; however, it is inefficient because it takes a long time. This leads to an unsatisfying experience for the passenger, but also for the travel company if the passenger goes elsewhere or complains about the delay caused by the system.

What is the *context* of use? Where and how is the system being used? Is the system usable in the different environments in which it is expected to be used? For example, some laptops have backlit keyboards so that they can be used in lower lighting conditions.

Other usability issues include **learnability** – how easy it is for users to accomplish basic tasks the first time they encounter the UI – and **recoverability** – how easy it is for the user to recover from errors. It is important to distinguish usability from usefulness or functionality, which is concerned with the capabilities and features of a system and has no bearing on whether users are able to use them or not.

Activity 6.15 (exploratory)

Many people have forms of colour vision deficiency, or ‘colour blindness’. A common version of this is ‘red–green’ colour vision deficiency that affects around 1 in 12 men and 1 in 200 women. People with colour blindness have difficulty in distinguishing between shades of red, yellow and green.

How might this affect the design of an interface that presents lots of information and options for a user?

Comment

The main point to consider is to check that colour is not the only way that information is being conveyed on the interface. An interface that presents a lot of detailed information in the form of charts, maps or tables should not present similar tones of red and green too close together if it is important to distinguish between the different elements. It may be necessary to include extra ways of separating details, for example, including symbols or different kinds of dotted lines.

This also applies to buttons and icons for different kinds of actions, and you might consider designing these with different sizes and shapes, in addition to using different colours.

6.5.2 Accessibility

One of the goals of HCI and interface design is **universal usability**, which refers to the design of information and communications devices, applications and services that are usable by all people. Universal usability or the notion of 'usable by all' is closely related to two other ideas: universal design or 'designed for all', and universal accessibility or 'accessible by all'. But what is **accessibility**?

Put simply, accessibility means having equal access to products, information and services regardless of people's physical or developmental abilities or impairments. This means that people with different abilities can perceive, understand, navigate and interact with computer systems. A key point to remember is that accessibility is about considering the wide spectrum of human physical and cognitive abilities. These different abilities can sometimes be temporary, for example, difficulty in using a keyboard due to a broken arm.

It is important to appreciate that if something is universally usable, it will be universally accessible. This means that some products designed specifically for disabled users can be beneficial to all users. Consider, for example, ramps and dropped kerbs: these have been added to pavements and buildings to accommodate wheelchair users, but their benefits extend to pushchair users, delivery service workers, cyclists and people pulling wheeled luggage.

Activity 6.16 (self-assessment)

Give one or two other examples of products or services that have been designed to be universally accessible.

In the same way that it is important to consider the impact of universal usability on our physical world, we should also consider its impact on the world of web-based, mobile and ubiquitous computing. For example, in a web context, features such as text descriptions, captioned videos, labelled forms and tables benefit us by enhancing web access. By implementing measures to improve web accessibility, we can all enjoy the same freedom of access to the web, regardless of the browsing technology we use. Examples of **assistive technology** enabling users with disabilities to gain access to the web, some of which are shown in Figure 6.22, include:

- screen magnifiers for viewing web pages at a larger size

- screen readers for hearing information on pages
- refreshable braille displays to read pages by touch
- keyboards with larger keys than standard ones
- mouse substitutes such as joysticks and trackballs
- speech recognition or voice recognition applications to replace keyboard input
- eye-operated communication and control systems, as shown in Figure 6.23.

(a) (b) (c)

Figure 6.22 Input and output devices: (a) trackball, (b) joystick and (c) refreshable braille display

Figure 6.23 A student using the Eyegaze eye-tracking system

There are a number of basic, well-established principles for accessibility that apply to all computer-mediated products and applications.

- Allow for user customisation – particularly of text size and style, background and foreground colours.
- Provide equivalent visual and auditory content and interface elements – text descriptions for images and video, transcription of auditory content, text labelling of interface elements.
- Provide compatibility with assistive technologies such as screen readers, screen magnifiers and speech recognition software.
- Follow operating system conventions.

- Allow access to all functionality from the keyboard alone – do not require the use of a mouse.
- Support efficient navigation by providing context and orientation information, and make this information available to assistive technologies.

Activity 6.17 (exploratory)

This activity involves using an assistive technology provided with your computer's operating system to access some information on the web. It is designed to encourage you to reflect on the importance of HCI and interface issues for accessibility. You will find this activity on the module website.

The main problem with screen readers seems to be that information can only be read out if the computer recognises it as text; documents or web pages that use graphics or photographs without alt text cannot be read. Alternative text or alt text is used for accessibility purposes to describe images in screen readers. Tooltips are used to give information about on-screen objects such as menu items and command buttons in conjunction with the cursor. The user points the cursor over an item, without clicking it, and a small hover box appears containing information about the item.

In Section 6.2.2, you learned that there are problems with the notion of 'the user', and that it is preferable to think instead in terms of 'a user' or 'users', or to identify specific types of users. This position is reinforced when we consider issues of accessibility and the need for assistive technologies to provide people with disabilities with access to products, applications and services through the provision of alternative interfaces. Designing interfaces to be accessible is a step towards inclusivity and bridging the 'digital divide', which you will study later in this module.

6.5.3 Evaluation with users

You have probably realised by now that usability is an important subject covering a whole range of issues. For example, there are many methods drawn from a variety of disciplines that are currently used to evaluate usability, and various approaches and techniques used to collect both qualitative and quantitative data from users. I will look only briefly at three of these techniques here, since an exhaustive study of the topic is beyond the scope of this module.

In Section 6.4.4, you learned that one of the principles on which UCD is based is **iterative** design, which involves researching user requirements and then following a design, build, evaluate and then redesign loop. **User evaluation** is a critical activity within the design process since it allows feedback to be incorporated so as to improve the design of the system. The following methods are often used at the start of a design or redesign process in order to establish initial requirements, as well as at stages during the iterative design process. While it is beyond the scope of this module to examine how to evaluate with users in detail, it is worth briefly mentioning some of the main techniques involved.

User observation

User observation involves watching what users do and listening to what they say when carrying out tasks. Sometimes use is made of measuring and recording equipment, including audio and video, in order to time the performance of tasks and record the number of errors made by users as they interact with the interface of a system. In this way, both informal and descriptive – or *qualitative* – and formal and numerical – or *quantitative* – empirical data can be gathered about user interaction.

User observation sessions are carried out either in the same environment in which the users will actually use the system – what is known as a *field study* – or in a specially designed user-evaluation laboratory – what is known as a *controlled* study.

Interviews

Interviews involve asking users a series of questions, which can be closed, open-ended or a combination of the two, and making a note of their responses. Interviews can be conducted both before and after users have interacted with a system: the former is useful for gaining an understanding of the expectations of users as well as for gathering knowledge about their skills and backgrounds, while the latter is useful for collating their impressions of the system.

Questionnaires

As with interviews, questionnaires can be distributed to users both before and after they have interacted with a system. They offer less scope for exploration and obtaining open-ended feedback from users than interviews on account of the fact that they are usually completed in the absence of the person who constructed them (i.e. the evaluator). However, when properly designed, questionnaires can prove extremely useful for gathering quantitative data that can be analysed statistically. This is particularly the case when use is made of metrics such as the **Likert scale** to rank performance, as shown in Figure 6.24.

How would you rate your overall satisfaction with the interface?

Very Satisfied	Somewhat Satisfied	Undecided	Somewhat Dissatisfied	Very Dissatisfied
5	4	3	2	1

Figure 6.24 A Likert scale used to rank user satisfaction with an interface

Activity 6.18 (exploratory)

In this activity you will be using your knowledge of HCI to evaluate the user interfaces of two websites. You will find this activity on the module website.

6.5.4 Conclusion

In this section, you studied usability and accessibility, and learned the basics of user evaluation. You also considered the international standard that is used to distinguish good usability from poor usability. Finally, you consolidated what you learned by reflecting on how these methods and techniques can be used in practice.

Note: If you are interested in studying this topic further, you might want to look up some of the leading figures in this area such as Jakob Nielsen, Jonathan Lazar and Ben Shneiderman.

6.6 Employability

You will find this section on the module website.

Summary

In this part, you studied a range of issues to do with 'crossing boundaries' for computer-based devices so that people can interact with them successfully. You have seen that it is important to choose the right interface for technology, because the consequences of not doing so could be costly and possibly even disastrous. You have seen that interfaces allow a user's model of the task to be separated from the builder or programmer's model of the underlying system.

Technology should be matched to the person, not the other way round. Users should be at the centre of the design process. There is no such thing as 'the user'; rather, there is 'a user' or 'users'. Therefore, different interfaces may be needed for different users. Mismatch between user and programmer models can create problems, which are the responsibility of the interface designer to attempt to resolve through the use of HCI techniques.

After studying this part, you should be able to:

- explain the meaning of the term 'human–computer interaction' (HCI) and its importance
- explain the meaning of key terms in HCI
- explain what design is and why it is important
- explore some of the reasons a product may succeed or fail in the marketplace
- reflect on the role of user interfaces in the success or failure of a computer system
- apply design principles of HCI in simple situations
- evaluate an interface using one or more user-evaluation techniques.

Finally, you should have identified some of the employability skills that you are acquiring through your study of TM111.

Answers to self-assessment activities

Activity 6.4

Put simply, the widespread use of the QWERTY layout by large numbers of people created inertia – a resistance to change that, in turn, afforded the QWERTY layout a momentum that allowed it to impose itself on later technologies.

Activity 6.7

You can be forgiven for thinking that this was a tough question. The average user will probably answer (1); a programmer will typically answer (2). This is a common example of the difference between how users think about a system and how programmers and designers do. Importantly, there is no right answer to this question. For example, programmers might argue that if you don't get a list of the documents in the folder, how could you go about getting such a list printed? Conversely, users might argue that this is inconsistent: if printing a document prints its contents, printing a folder should print its contents too. The average user often can't say why they think this is natural; they simply know it is.

Activity 6.8

Human–computer interaction, as a discipline, is the study of how people interact with computer systems. It is concerned with designing systems so that they are safe, efficient, easy and enjoyable to use, as well as functional.

The interface, or user interface (UI), is that part of a computer system with which users interact directly.

Separability refers to how an interface separates what lies on either side of it so that changes can be made on one side without affecting the other.

Activity 6.13

I thought of the ring tone and beep of a mobile phone to indicate an incoming call and a text message, respectively. You are probably also familiar with the warning sound generated by the operating system running on your computer when you attempt to close a file you have just been editing without first saving it.

Activity 6.16

I thought of automatic doors that open when pressure is exerted on sensors built into the floor, and lift doors.

There are also the subtitles provided on some films and TV drama. I've often switched those on when I'm struggling with different accents or actors who speak fast/mumble.

Some taps are operated with a lever arm rather than screw; likewise the caps on some lotion tubes that have a squarish rather than round profile which are much easier to operate even for people with good manual dexterity.

There is also the automatic sound warning given when large vehicles are reversing.

References

Grudin, J. (1993) 'Interface: an evolving concept', *Communications of the ACM*, vol. 36, no. 4, pp. 110–19.

ISO (1998) ISO 9241: *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)*, 'Part 11: Guidance on usability', Geneva, International Organization for Standardization.

Norman, D. A. (1988) *The Design of Everyday Things*, New York, Doubleday.

Acknowledgements

The module team acknowledges the contribution of Mustafa Ali and Nick Dalton to the production of this part.

Grateful acknowledgement is made to the following sources.

Figure 6.1(a) and 61(b). © Visualfield/iStockphoto

Figure 6.3(c), (d) and (e). Taken from Wikipedia. / This file is licensed under the Creative Commons Attribution-Noncommercial-ShareAlike Licence http://creativecommons.org/licenses/by-sa/3.0/

Figure 6.3(i). © www.blackberry.com

Figure 6.8. © www.microsoft.com

Figure 6.13. Taken from https://lifehacker.com/how-predictive-keyboards-work-and-how-you-can-train-yo-1643795640

Figure 6.14: © David Moore Lifestyle / Alamy Images

Figure 6.18: © AF Archive / Alamy Images

Figure 6.19(a). Taken from https://www.designlabthemes.com/principles-of-flat-design/

Figure 6.19(b). Taken from http://www.lifesomundane.net/2013_02_01_archive.html

Figure 6.22(a). http://upload.wikimedia.org/wikipedia/commons/9/96/Trackball-Kensington-ExpertMouse5.jpg, Creative Commons Licence.

Figure 6.22(b). CTI Electronics

Figure 6.22(c). Taken from Google Inc

Figure 6.23. © www.microsoft.com

Every effort has been made to contact copyright holders. If any have been inadvertently overlooked, the publishers will be pleased to make the necessary arrangements at the first opportunity.

Index

Entries and page references in **bold** refer to glossary items.