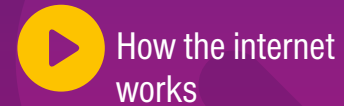


# Computer networks



## How do computers communicate?

Connecting computers to form computer networks and the internet (a network of networks) has had a huge impact on our lives.

Think about how limited our use of technology in school would be if we had no access to the local network or the internet. Think about how frustrating it is when we have no data signal for our smartphones or wifi for our laptops.

The internet has made it possible to communicate and collaborate with a richness and immediacy never experienced before. And yet, it's something that most of us take for granted.

The new computing curriculum sets out to change this. It requires that pupils are taught:

- in key stage 1: to 'recognise common uses of information technology beyond school'
- in key stage 2: to 'understand computer networks, including the internet [and] how they can provide multiple services, such as the world wide web'
- in key stage 2: to 'use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content'<sup>1</sup>

### How does the internet work?

The internet is a physical thing: it's the cables, fibre, transmitters, receivers, switches, **routers** (and all the rest of the **hardware**) that connects computers, or networks of computers, to one another.

The internet has been designed to do one job: to transport **data** from one computer to another. This information might be an email, the content of a web page, or the audio and video for a video call.

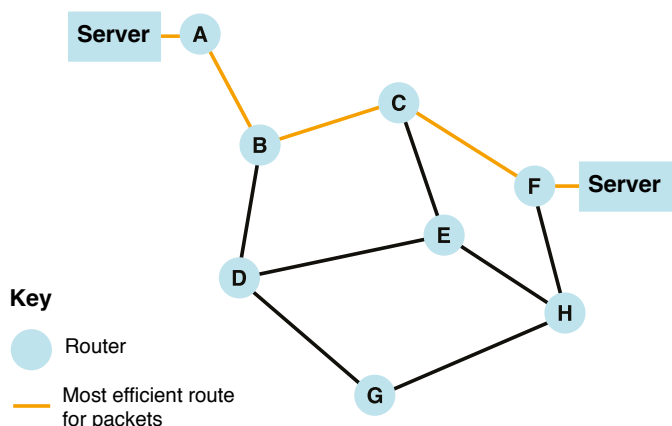
The data that travels via the internet is digital: this means it is expressed as numbers. All information on the internet is expressed this way, including text, images and audio. These numbers are communicated using binary code, which is made up of 1s and 0s, using on/off (or low and high) electrical or optical signals. Binary code is similar to the Morse code used for the telegraph in Victorian times, but it's much, much faster. A good telegraph operator could work at maybe 70 characters (letters) a second, but even a basic school network can pass data at 100 million on/off pulses a second, enough for some 12.5 million characters per second. One transatlantic fibre connection has the capacity for up to 400 billion characters per second!

Digitised information needs to be broken down into small chunks by the computer before it can be sent efficiently. These smaller chunks of data are known as 'packets'.

The small **packets** can be passed quickly through the internet to the receiving computer where they are re-assembled into the original data. The process happens so quickly that high definition video can be watched this way, normally without any glitches.

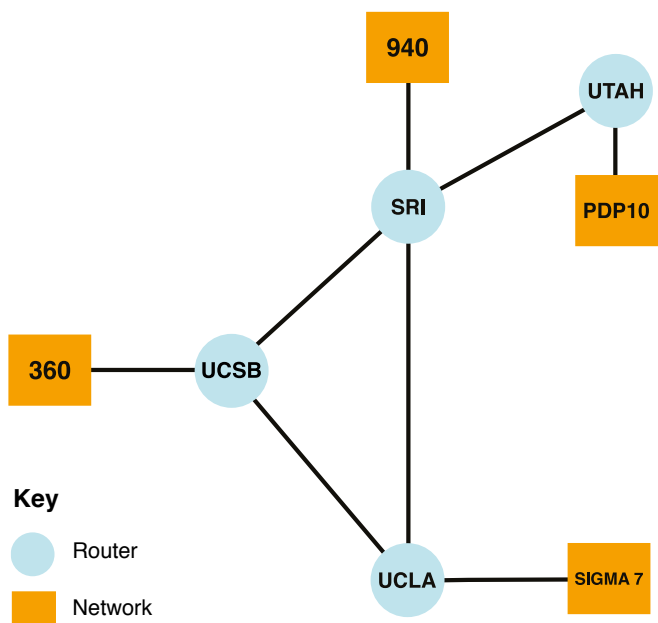
<sup>1</sup> National Curriculum in England, *Computing Programmes of Study* (Department for Education, 2013).

The packets don't all have to travel the same way through the internet: they can take any route from sender to recipient. However, there is generally a most efficient route, which all the packets would take.



A sample network: note there is more than one route for packets to travel.

It's perhaps easier to understand how the internet works now by looking at a picture of how it worked in 1969 when it started:



- SRI** Stanford Research Institute
- UTAH** University of UTAH
- UCSB** University of California, Santa Barbara
- UCLA** University of California, Los Angeles

Here you see the internet made up of just four routers: UCLA, SRI, UCSB and UTAH. Each router is a piece of hardware that passes packets of data from the networks they are connected to (in the case of UTAH, PDP10, in the case of UCLA, SIGMA 7) to any of the other three networks.

So if you were using the PDP10 network at the University of UTAH and sent a message to someone

at UCLA, your message would be passed first to your router at UTAH, then on to the router at Stanford Research Institute (SRI), then (normally) to UCLA's router, where it would be passed on to whichever recipient it was intended for on their SIGMA 7.

The internet is obviously much, much bigger than this example. In real life, the journey of a packet of data from your home computer to one of Microsoft's server farms might look something like this:



When you type a URL (such as [www.bbc.co.uk](http://www.bbc.co.uk) or [www.computingatschool.org.uk](http://www.computingatschool.org.uk)) into your browser you send a packet of data requesting the content of these pages to be returned to you. But before this can happen, the domain name first needs to be converted into numbers. This is the job of the **Domain Name Service (DNS)**, which converts these familiar web addresses into numbers known as **IP (Internet Protocol) addresses**. The DNS itself uses the internet to look up (in the equivalent of huge phone books) the numeric address corresponding to the domain names.

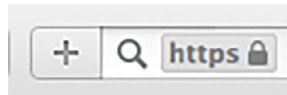
Each packet has a destination IP address on it. With it the router can easily look up which way to pass the packet on.

## Who can see the data we transmit?

There's nothing to stop routers from looking at the data in the packet before they pass it on (just as there was nothing to stop telegraph clerks reading the messages they passed on in Morse code).

To be able to send information, such as passwords or bank account details, secretly via the internet, it's important to **encrypt** the data first. This happens automatically when using the 'https' version of websites (see page 37). In these situations, you'll see a little green padlock displayed in your browser's address bar.

The data is decrypted when it reaches its destination.



## Classroom activity ideas

- Ask pupils to draw a picture of the internet. This will allow you to spot any misconceptions they have, and provide an opportunity for pupils to share their understanding.
- Carry out this 'unplugged' activity to model how the internet passes packets of data.
  - » Organise all but four of your pupils into groups.
  - » Tell the pupils to choose one pupil in their group to be the 'group router'. The rest of the group will be 'computers'.
  - » Ask the remaining four pupils to take on the role of 'internet routers', which connect the group routers together.
  - » Give each 'computer' a numerical address, comprising a group number and a computer number (e.g. 1.1, 1.2, 1.3; 2.1, 2.2, 2.3, etc.).
  - » Ask each 'computer' to write a short message to another 'computer' in a different group, splitting their message over three different slips of paper and marking their slips '1 of 3', '2 of 3' and '3 of 3'. Tell them to write their numerical address and the numerical address of the recipient, e.g. 'To: 2.2; From 3.4; 2 of 3.' This is the 'packet header'.
  - » Ask the 'computers' to pass their slips to their 'group router', who can pass these on one at a time to the 'internet routers'. They in turn pass them to the correct 'group router' who passes them to the recipient themselves, who can reassemble the message as their other packets arrive.

Table 1 (6 pupils)

Router 1.0	1.1	1.2
1.3	1.4	1.5

Table 2 (6 pupils)

Router 2.0	2.1	2.2
2.3	2.4	2.5

Message slips

To: 2.3  
Sequence: 1 of 3  
Data: What  
From: 1.5

To: 2.3  
Sequence: 2 of 3  
Data: is for  
From: 1.5

To: 2.3  
Sequence: 3 of 3  
Data: tea?  
From: 1.5

Role-playing a computer network in class.

- Investigate the physical infrastructure of the school network. Tell the pupils to walk from their laptop to the local wifi point, or to follow the network cable from the computer to the classroom switch. Next, walk together to the school's main network switch, firewall and router. If you can, then walk down to the nearest BT green cabinet, and perhaps to your local telephone exchange, depending on how close this is to you.
- Explore the steps on the journey of a packet using the 'tracert' command at the Windows command prompt, if you have access to this. Also see the Visual traceroute reference in Further resources.
- Ask your school network manager to talk pupils through how the school network connects their computers to the rest of the internet.



## Further resources

- Bagge, P., 'Year 5 Computer Science Planning', available at: [www.code-it.co.uk/year5/index.htm](http://www.code-it.co.uk/year5/index.htm).
- Barefoot on 'KS2 Modelling the Internet activity', available at: <http://barefootcas.org.uk/programme-of-study/understand-computer-networks-including-internet/ks2-activity-modelling-the-internet/> (free, but registration required).
- Barefoot on 'Internet Services', available at: <http://barefootcas.org.uk/programme-of-study/multiple-services-provided-networks-internet/internet-services/> (free, but registration required).
- BBC Bitesize clip 'Computer networks – LAN and WAN', available at: [www.bbc.co.uk/learningzone/clips/computer-networks-lan-and-wan/4381.html](http://www.bbc.co.uk/learningzone/clips/computer-networks-lan-and-wan/4381.html).
- Blum, A., *Tubes: Behind the Scenes at the Internet* (Viking, 2012).

- Andrew Blum’s talk ‘Discover the physical side of the internet’, available at: [www.ted.com/talks/andrew\\_blum\\_what\\_is\\_the\\_Internet\\_really](http://www.ted.com/talks/andrew_blum_what_is_the_Internet_really).
- Artistic representations of what the internet means, available at: [www.canyoudrawtheinternet.com](http://www.canyoudrawtheinternet.com).
- Mark Dorling and others’ Digital Schoolhouse planning on networks, ‘Networks Unplugged’, available at: [www.digitalschoolhouse.org.uk/documents/networks-unplugged-workshop-pack](http://www.digitalschoolhouse.org.uk/documents/networks-unplugged-workshop-pack).
- Naughton, J., *From Gutenberg to Zuckerberg: What You Really Need to Know About the Internet* (Quercus, 2012).
- Visual traceroute to find the path from their web server to an internet address, available at: [www.yougetsignal.com/tools/visual-traceroute/](http://www.yougetsignal.com/tools/visual-traceroute/).

### What can you do with the internet?

Picture the train network, efficiently routing trains of all kinds from one point to another, irrespective of what those trains contain. Some will have passengers, others freight, others are perhaps maintenance stock. In the same way, the infrastructure of the internet can be used for lots of different things.

The services which run on computer networks, including the internet, fall into roughly two groups:

1. client–server: one computer (the client) accesses services or content running or stored on another, typically larger, computer (the server)
2. peer-to-peer: two computers communicate directly as equals, passing data directly to and from each other.

#### Client–server



Client computer



Server computer

#### Peer-to-peer



Client computer



Client computer

The World Wide Web (see page 36) fits into the client–server model, but so do lots of other services which use computer networks and the internet as a means of communicating.

A school network will often have one or more computers acting as servers, responding to requests from the desktop, laptop and tablet computers which act as clients. On a local area network (LAN) like this, the servers might provide: central storage and backup for files, access to documents, etc. from any computer on the network, a management information system (such as SIMS), local email accounts, access to printers, username and password authentication, filtering and logging of access to the web and even locally stored copies of frequently visited web pages.

Email is a good example of a client–server system using the internet (although many people’s experience of email is as webmail accessed through a browser like Internet Explorer). The journey of an email might be something like this:

- Alice opens up Outlook and starts typing in her email to Bob. She includes Bob’s email address, bob@builders.com, in the ‘To’ line of the email and clicks ‘send’.
- The email is transmitted via the internet (or the local network) to her outgoing mail server. If the email is intended for another domain (builders.com here) rather than Alice’s own (lookingglass.org) then Exchange will forward the email as packets of data via the internet, which routes these through to the incoming mail server for builders.com as discussed above.
- The inbound mail server at builders.com (again perhaps running Exchange) re-assembles the message from the packets of data, accepts this and stores this ready for Bob to collect.
- Later on, Bob’s email client (perhaps also Outlook) connects to his mail server and asks if there are any messages for him. The one from Alice gets transmitted to Bob’s computer via the local network or the internet, where Bob can read it in his email software.

Although it might look to Alice and Bob as though they are communicating directly with each other, all their emails are going via the outbound and inbound mail servers. Notice that the contents of their emails aren’t encrypted, so the organisations running the two mail servers can read the contents of these messages if they wish.

Not all communication on the internet uses a client–server model. For example, peer-to-peer communication is a model used for Skype and a number of other video conferencing or voice over internet systems. Although Skype uses a server to maintain a list of logged-in users and the IP address of their computers, when a call is

connected the packets of data that make up the digitised video and audio for the call are routed directly through the internet between the two parties.

Some online gaming websites use a similar peer-to-peer system, as does BitTorrent, a protocol which allows large files to be shared between many computers by allowing direct peer-to-peer connections. Because peer-to-peer connections are harder for large organisations to monitor, they are favoured by those using the internet for criminal purposes, for example the use of the BitTorrent protocol for illegally sharing copyrighted material.



### Classroom activity ideas

- Role-play can be used very effectively to teach how email works and issues with email security. Explain to pupils that email addresses can be 'spoofed' or accounts hacked. So, not all emails are from who they appear to be. Warn them that files attached to emails can contain viruses. Also explain that links in emails can sometimes point to websites that are set up to capture personal information such as passwords. You might like to run this as part of a larger topic looking at the effective and safe use of email, perhaps in a twinning project with a class in this or another country.
- Share and write a range of emails and written letters. Discuss the advantages and disadvantages of each type of communication.
- Use a video conferencing system to allow experts to talk to the class or to allow two classes to communicate. As you set up the computer, talk through the technical aspects of the call with your pupils. Note: Skype and most other video conferencing systems don't allow children to register for accounts, so you will need to run this as a whole-class activity.
- Encourage pupils to talk about how they and their families use the internet to communicate, highlighting any services they use in addition to the World Wide Web.

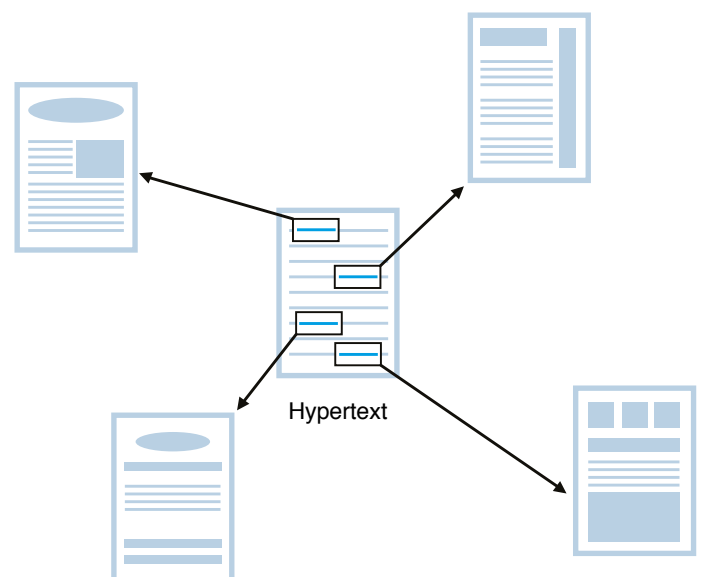


### Further resources

- Guha, S., Daswani, N. and Jain, R., 'An Experimental Study of the Skype Peer-to-Peer VoIP System' (2006), available at: <http://saikat.guha.cc/pub/iptps06-skype.pdf>.
- The journey of a letter, available at: [www.anpost.ie/anpost/schoolbag/primary/our+people/the+journey+of+your+mail/](http://www.anpost.ie/anpost/schoolbag/primary/our+people/the+journey+of+your+mail/).
- 'Story of Send on Google Green' (a short cartoon about the journey of a gmail), available at: [www.youtube.com/watch?v=5Be2YnlRIg8](http://www.youtube.com/watch?v=5Be2YnlRIg8).

### What is the World Wide Web?

In 1989, British computer scientist Tim Berners-Lee decided to combine the capabilities of the internet with the functions of hypertext (documents that include hyperlinks that allow connections to be made between different files) to manage information systems at CERN where he was working.



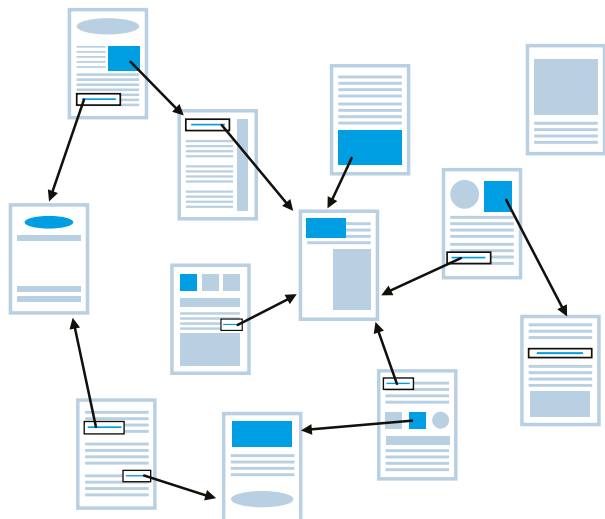
The links in the hypertext take the reader to different documents which extend or support the information in the original document.

Berners-Lee developed a specification for how an internet-based version of hypertext would work and then wrote the software for the first **web servers** and web browsers. The result was the **World Wide Web**.

The internet is about connecting computers together, but the World Wide Web is about the connections between documents. When you click on a web link, another web page is requested from (typically) a different web server somewhere else on the internet.



The content of this web page is then delivered to your web browser.



The World Wide Web is about the connection (the links) between documents.

*What standards does the World Wide Web use?*

To ensure that all computers could communicate with one another, Berners-Lee developed a set of standards (called protocols) for the Web. Versions of these are all still used today.

## 1. HTTP (HyperText Transfer Protocol)

This is the process that computers use to request and transfer hypertext to one another.

The Web is a client–server system: we use a web browser on our computer to request a web page from one of the many, many web servers connected to the internet. The request travels as a packet of data via switches and routers until it reaches the intended web server. The server responds by sending back the content of the page, together with any images and formatting instructions and mini programs (typically in JavaScript) needed for the page. If the page isn't there, it sends back a '404: Not found' error message – sometimes you'll see other error messages too.

Remember that the internet doesn't encrypt packets of data: there's another version of HTTP, called HTTPS, where the request for a page, the contents of the page and any information entered into a form (such as a password) are sent over the internet in an encrypted form. This encryption can sometimes be bypassed by network managers and government agencies.

## 2. URL (Uniform Resource Locator)

URLs are the precise location on the Web where web pages or their components are stored. It's what you type in to your browser's address bar to request a page.

Each bit of a URL means something. Let's look at the URL of one of the first web pages – Berners-Lee's home page for the World Wide Web project itself – to work out what each bit means:

<http://info.cern.ch/hypertext/WWW/TheProject.html>

- **http**: this is the protocol we're using to request hypertext and the content that comes back – see above.
- **://** is just punctuation – Berners-Lee now thinks it would have been better if he'd skipped the // bit!
- **info** is the name of the web server we're connecting to. Often this will be www these days, or this is just omitted as the main web server for the organisation will be assumed.
- **cern** is the name of the organisation, in this case the European Centre for Nuclear Research.
- **ch** is an abbreviation for the country where the organisation has registered their domain name, in this case Switzerland. Some countries also show what sort of organisation it is registered as, e.g .co.uk for a commercial site and .sch.uk for a school site in the UK. If no country is shown, then it will be registered in the USA: .com for commercial sites, .edu for university sites, and so on.
- **/hypertext** is a directory (folder) on the web server.
- **/WWW** is a directory inside the /hypertext directory on the web server.
- **TheProject** is the name of the actual file we're requesting, in this case a web page about the World Wide Web project. Sometimes you don't see a file name at the end of a URL, in which case the web server will send back the default file for the directory, often an index page such as index.html.
- **.html** is the file extension, which shows what format the page is written in, in this case **HTML** (see page 38). This is like .doc or .docx for a Word file, or .jpg or .jpeg for an image.

Although it is often convenient to use search engines like Google or Bing to find pages rather than typing in URLs, the URL is a good way to check that you're connecting to the web server you think you are (rather than a spoof website). URLs are also useful when acknowledging sources of information, and for creating links between pages (and so building more of the connections that make the Web so useful).

### 3. HTML (HyperText Mark-up Language)

HTML is the computer language (code) in which the content and structure of a web page are described or 'marked up'.

The content of web pages is stored in HTML format on web servers. Creating a web page involves writing (or getting a computer to generate) the HTML that describes the page. HTML can be read, and written, by humans as well as computers. You can view the HTML source code for any web page using tools built into your web browser. (There's a menu command to do this, or you can press 'ctrl-u' in Internet Explorer.)

These days, the HTML for a web page might not be stored as a file on the web server: in content management systems, when a page is requested it will be generated automatically using a database of content, a template and some programs running on the web server. For example, every time you visit [www.bbc.co.uk/newsround/](http://www.bbc.co.uk/newsround/) the page will be generated using the latest news in the database.

More recently, a couple of other languages have come to play an important part in developing for the Web.

#### CSS (Cascading Style Sheets)

CSS provides formatting information alongside the content and structure of HTML, allowing designers and developers to specify exactly how the content of the page should be displayed in the web browser on a computer, tablet, smartphone or printer.

#### JavaScript

JavaScript is a programming language that can be interpreted by the web browser itself, allowing interaction with the content of a page to be handled by the user's computer (the client) rather than on the server itself. The web-based version of Office 365 relies heavily on JavaScript.

*What's the most amazing thing about the Web?*

The amazing thing about the Web isn't really these technologies though. It's that, from its early days as the preserve of academic scientists, so many organisations and individuals have connected their own web servers to the internet and added their own content to the Web. In part this was because Berners-Lee created a system that was accessible, scalable and extensible, capturing the imagination of many. But it's also because he and

CERN gave it to the world for free – the standards and the technology were entirely open, without any central authority or commercial company licensing or charging for their use.



#### Classroom activity ideas

- The national curriculum for history suggests that key stage 1 pupils could look at William Caxton (who introduced the printing press to England in the fifteenth century) and Tim Berners-Lee as examples of 'the lives of significant individuals in the past who have contributed to national and international achievements'. Compare the life, work and influence of these two figures.
- Encourage pupils to look at the different parts of the URLs for the web pages they visit, asking them to explain what each part of the URL means. Make a display showing the different parts of some interesting or common URLs.
- Ask pupils to talk to their parents, grandparents or carers about the difference the World Wide Web has made in their lives.
- Tell pupils to keep a diary of the different ways they use the Web over a week.



#### Further resources

- BBC Bitesize 'What is the world wide web?', available at: <http://www.bbc.co.uk/guides/z2nbgk7>.
- Tim Berners-Lee 'Answers for Young People', available at: [www.w3.org/People/Berners-Lee/Kids.html](http://www.w3.org/People/Berners-Lee/Kids.html).
- The original CERN home page for the Web, available at: <http://info.cern.ch/hypertext/WWW/TheProject.html>.
- Codecademy curriculum materials, available at: [www.codecademy.com/schools/curriculum](http://www.codecademy.com/schools/curriculum) (registration required).
- Mozilla Web Literacy whitepaper, available at: <http://mozilla.github.io/webmaker-whitepaper/>.
- Wayback Machine to search for historic web pages, available at: <http://archive.org/web/>.

*How do you make a web page?*

There are plenty of tools available for you and your pupils to create your own content for the Web.

Your school's learning platform or VLE provides one way to get content online, as do blogging platforms

like WordPress. These platforms usually include a ‘WYSIWYG’ (what you see is what you get) editor. This makes writing content for the Web similar to using Microsoft Word, with a range of formatting controls built in. In most of these editors, you can swap into code (or source view), seeing and editing the HTML itself. This can be a good introduction to working directly in HTML, as you can always swap back to the WYSIWYG view to see the effects of editing the code.

Giving pupils some experience of writing content for the Web through editing HTML ‘by hand’ is well worth doing although it isn’t, strictly speaking, programming. It adds to their understanding of networks including the internet that the national curriculum at key stage 2 expects, and is one more way of using software on a range of devices to create content. It is also a good way to get pupils used to working in a formal, text-based computer language. As with other text-based languages, working in HTML helps reinforce the importance of spelling, punctuation and grammar: mistakes in the mark-up of the page usually become quite apparent in the way the browser displays the page.

Many pupils are likely to find these skills useful in the long term too, both at secondary school and beyond: developing content for the Web is part of many jobs, teaching included.

### What does HTML look like?

Let’s compare the HTML code for a simple web page and the page itself.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>A simple webpage</title>
  </head>
  <body>
    <h1>Origins of the Web</h1>
    <p>Tim Berners-Lee started working on
the world-wide web project in 1989.</p>
    <p>He was working at <a href="http://
home.web.cern.ch/">CERN</a> in Switzerland
at the time.</p>
    
  </body>
</html>
```

## Origins of the Web

Tim Berners-Lee started working on the world-wide web project in 1989.

He was working at [CERN](#) in Switzerland at the time.



Can you see where the content for the page comes from in the code? Can you see what effect some of the HTML tags (the bits in the < . . . > angle brackets like <h1> and <p>) have on how the content is structured?

Notice how most of the tags come in matched pairs, e.g.

- <html> and ending </html> for the whole page
- <head> to </head> for the information about the page, such as its character set and title
- <body> to </body> for the content of the page
- <h1> to </h1> around the main heading for the page
- <p> to </p> around each paragraph.

Compare the underlined link in the web page with the corresponding code. In the code, <a> to </a> show where the link should be and href=“http://home.web.cern.ch/” inside the <a> tag detail where the link should point to.

An image is inserted from elsewhere on the web, using a single <img> tag, this time without a matched closing tag, and again giving the location of the image using src=“http://upload.wikimedia.org/wikipedia/commons/thumb/7/7e/Tim\_Berners-Lee\_CP\_2.jpg/320px-Tim\_Berners-Lee\_CP\_2.jpg” inside the <img> tag.

### How do I get started with HTML?

Mozilla’s Thimble tool for creating websites (available at: <https://thimble.webmaker.org/>) makes it easy to get started with coding in HTML, as it displays the source code alongside the resulting web page.



Rather than starting from a blank page, pupils can try editing other web pages, exploring the structure and HTML code of these pages and seeing what effect changing the code has on how the page is displayed in the browser.

On Internet Explorer, you can use the Developer Tools (hit F12, or launch via the menu) to view and edit the source code (the HTML code which describes the content and structure) for a page. Alternatively, you can install Mozilla's X-Ray Goggles as an active bookmarklet (see Further resources) to remix and share edited web pages.



### Classroom activity ideas

- When using their learning platform, VLE or class blog, encourage pupils to swap from the normal WYSIWYG (what you see is what you get) mode of the built-in editor into the code, source or HTML mode and try writing their post or page in that. Remind them that they can swap back and forth to see how the code relates to the page that's displayed. Give pupils a list of some common HTML tags to try out for themselves.
- Set pupils the challenge of making a parody of a web page by using either the Developer Tools in Internet Explorer or X-Ray Goggles to edit the code for the page. It's wise to decide some ground rules for this activity in advance. Show pupils how easily a spoof page can be created this way, and explain why it's so important to check the address of the page they're visiting to confirm it is authentic rather than merely one which looks convincing.
- Rather than asking pupils to write up a story or a report using Word, challenge them to do this using HTML code to make a web page. Emphasise that they need to concentrate on the content and structure of their page, which is what HTML is designed for. Encourage them to add in links to supporting material using the `<a>` tag if they're creating a non-fiction account, and perhaps to add in some images from elsewhere on the Web using the `<img>` tag.



### Further resources

- Learn to code tutorials from Codecademy, available at: [www.codecademy.com/](http://www.codecademy.com/) (registration required).
- Shay Howe 'Learn to Code and CSS' tutorials, available at: <http://learn.shayhowe.com/>.
- 'App Design Basics: Learn to code using HTML and CSS' from Playto, available at: <https://learn.playto.io/html-css/lesson/0>.

- Thimble: <https://thimble.webmaker.org/>.
- Tutorials on a wide range of computer languages from w3schools, available at: [www.w3schools.com/](http://www.w3schools.com/).
- See the source code behind web pages using X-Ray Goggles, available at: <https://goggles.webmaker.org/>.

### How does a search engine work?

Search engines like Google and Bing have transformed the way we use the Web. Instead of having to remember URLs for the pages we want, or following the links from one page to another, we can normally rely on these web-based programs to give us the most relevant results for our query.

Given how much we use search engines, it's important to use them effectively and efficiently, to show some discernment in deciding how far a particular page can be trusted, and to have some grasp of the **algorithms** that underpin them.

In order for Bing or Google to be able to respond to a search query, they use their index of the Web. A search engine builds its index by using specially written programs called 'web crawlers'. The web crawlers create a huge copy of the publicly accessible bits of the Web (called a cache) which is stored on the search engine's servers.

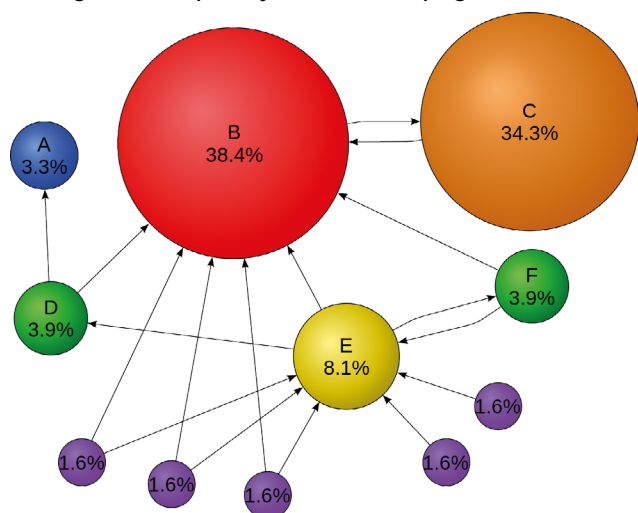
When a new or updated copy of a web page is added to the cache, an entry for the page will be added to, or updated in, the search engine's index of the Web for each of the words on the page (typically ignoring small, common words like 'and', 'the' and so on). The web crawlers continue to build and update the cache by following all the hyperlinks in the page, requesting and making copies of those pages too, adding or updating index entries for them, and following the links on those pages too. And so on.

So when we type in a keyword such as 'dog' into a search engine, it consults the index and returns a list of all the web pages on which that keyword appears. Typing in several keywords, e.g. 'dog' and 'bowl' would only return pages with both of these keywords, which helps to narrow down the set of results.

### How are search results ranked?

The really clever bit about web searches is not the list of results, but the right rank order the results are put into. How do the search engine algorithms decide what to put top of the list?

Google's founders, Larry Page and Sergei Brin, recognised that the key to determining how relevant a particular result was likely to be lay in the links between other pages and the result. They realised that a high quality page is a page that has lots of links *pointing to it* from other web pages, particularly if they too were high quality results. This is shown in the illustration below, where the larger the circle, the higher the quality of the web page.



The cached and indexed copy of the (publicly accessible) Web on the servers of search engines also includes the links between them. This allows Page and Brin's PageRank algorithm to work out which pages are considered the highest quality to other web developers (as they add links to those into their own content). Thus, for many queries the Wikipedia entry will often be at the top of, or at least high up, the results list, not because of its accuracy or authority, or even because people click on this more than other results, but because lots of the other high quality search results link to it.

The actual algorithms that search engines use can be very complicated and are frequently tweaked to keep one step ahead of the 'search engine optimisation' (SEO) industry that tries to improve the ranking for its clients' pages. These days, the ranking of results is typically personalised: based on location, the history of what the user's searched for and clicked on before, and close on 200 other factors or 'signals'.

When teaching pupils about how search engines work, point out the 'sponsored' results which are shown above or to the side of those generated using this relevance algorithm. The sponsored results are also algorithmically generated, based on the keyword, some quality measure for the advert, the page it points to and often your search history. They're placed on a 'pay per click' basis: the search engine doesn't charge for showing the advert, but the advertiser pays when you click on it, so it's in their interests to only show the most relevant adverts here.

The mechanics will vary from one search engine to another, but a good search engine should also: filter out explicit content automatically, allow you to search within a particular site, allow results to be filtered by their location (e.g. just the UK) and by date range (e.g. just pages created or edited in the last year). Some search engines even allow results to be filtered by reading level, for example restricting the results to those written using shorter words or less complex sentences.



## Classroom activity ideas

- Encourage pupils to use search engines for independent or guided research projects. Get pupils to experiment with the effect that adding in additional keywords or searching for phrases (by putting quotation marks around the phrase) has on a set of results.
- Demonstrate, and ask pupils to use, some of the more advanced search features, such as filtering by date and reading level. Show pupils how they can view the cached copy of a web page (for both Google and Bing this is hidden under the green drop-down next to the URL on the results page).
- Read through the Digital Schoolhouse notes on a simulation of how a search engine works, based on Google engineer Doug Aberdeen's presentation at the 2012 CAS Conference (see Further Resources below). Print off the resources and run this as an activity with your class.



## Further resources

- Doug Aberdeen's simulation from the CAS conference, available at: [www.computingschool.org.uk/index.php?id=aberdeen](http://www.computingschool.org.uk/index.php?id=aberdeen).
- Useful list of advanced search keywords in Bing, available at: <http://onlinehelp.microsoft.com/en-us/bing/ff808421.aspx>.
- Short animated presentation 'How Search Works' by Matt Cutts, available at: [www.youtube.com/watch?v=BNHR6IQJGZs](http://www.youtube.com/watch?v=BNHR6IQJGZs).
- Peter Dickman's lecture 'How Google Search Works', available at: [www.youtube.com/watch?v=C8v7AM1o7uM](http://www.youtube.com/watch?v=C8v7AM1o7uM).
- Digital Schoolhouse simulation of how a search engine works: <http://community.computingschool.org.uk/files/3874/original.pdf>.
- Eli Pariser's talk 'Beware online "filter bubbles"' (how individually focused our search results are), available at: [www.ted.com/talks/eli\\_pariser\\_beware\\_online\\_filter\\_bubbles?language=en](http://www.ted.com/talks/eli_pariser_beware_online_filter_bubbles?language=en).