ELT Computer Science

Sheriffo Ceesay
Nnamdi Ekwe-Ekwe

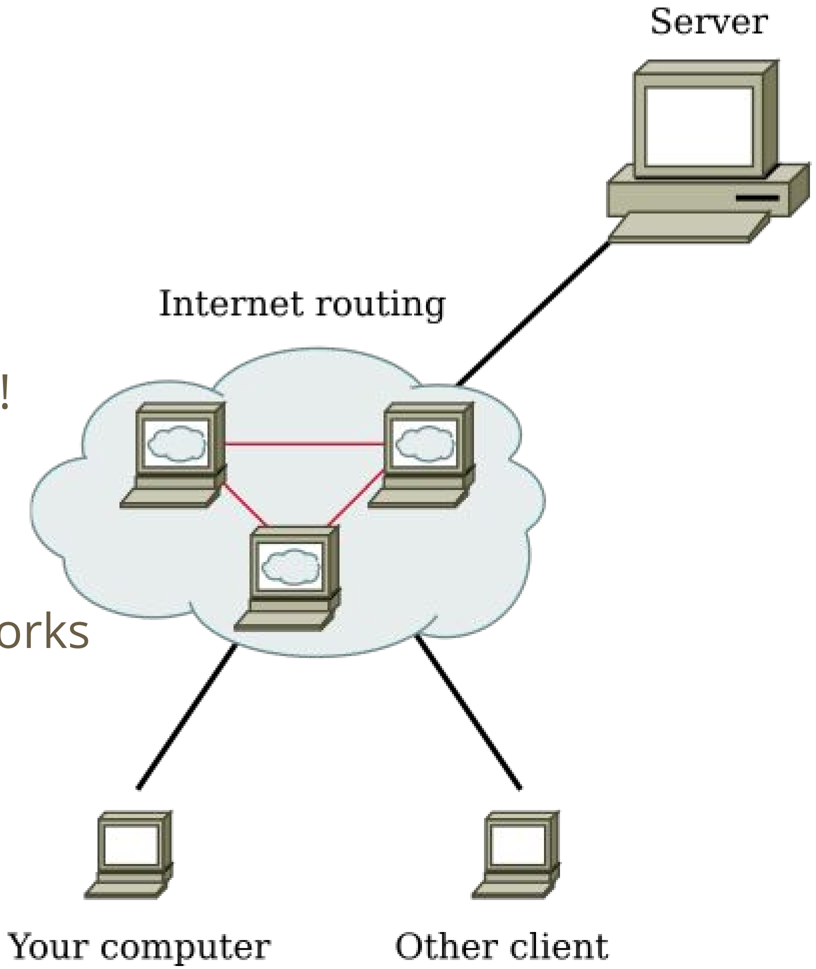
Greatest appreciation goes to Martin and Shyam for sharing this well formatted lecture materials.

What is a website?

Can be very complicated!

Simplest form: a computer with lots of files!

- Your browser asks for a file
- The file is sent in code
- The code describes how the website works
- Your browser receives the code
- Your browser draws the website



What code?

HTML

- HyperText Markup Language
- Describes what the page **is**
- Includes information about other files too
    - images
    - sounds
    - formatting (CSS)
    - behaviour (Javascript)

Viewing HTML

The Internet is **open source**

- If your browser uses a file, you can look at it
  - see how it was made
  - even copy and change it
- In your browser: **view source**
- Also: **view developer tools**
  - Play with both of these!

Server & client

Server: computer with all the files

- belongs to owner of website
- “serves” files to lots of people

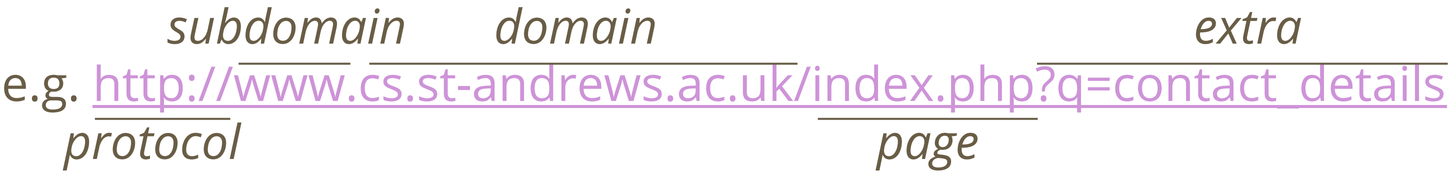
Client: computer asking for files

- yours!

We call this a “server-client architecture”

Finding a server

To ask for a file, we must provide a **URL**



- e.g. http://www.cs.st-andrews.ac.uk/index.php?q=contact_details

- *protocol*: what kind of response we expect
- *domain*: the website we want to access
- *subdomain*: the area of the website
- *page*: the exact file we want
- *extra*: any more information needed

What actually happens?

When you type the URL into your browser:

- browser searches for *domain*
  - asks a service called **DNS**: Domain Name Service
  - returns an **IP address** indicating a single computer
- browser sends its request (URL) to that IP address
  - using systems like for physical post
- server at that address receives request
  - finds/generates the information requested
- server sends back HTML
  - and possibly other files too

Default pages

http://www.cs.st-andrews.ac.uk still works

- but what page is being requested?

- Convention: **index** page is loaded first
  - index.html
  - index.php

- Try going to http://www.cs.st-andrews.ac.uk/index.php
  - exactly the same!

How complicated?

Websites can be complicated or simple

Static: server has a list of pages

- client requests the file as it is
- all clients see the same version of the file
- we will *only* investigate these this semester

Dynamic: server has instructions for creating pages

- client’s request can be broad or specific
- server creates page *for that client, at that time*
- e.g. Facebook: my feed now ≠ your feed now ≠ my feed yesterday

Doing it yourself

- Get a server
    - usually hire some space on someone else's
    - may also need a domain name
- Put files on it
- Tell people about it!

We will skip step 1 for now

- Time to create some files!

A basic HTML file

```
<head>
	<title>My first file</title>
</head>
<body>
	<h1>Hello world</h1>
</body>
```

- create a file called *index.html* with the above contents
- open with a browser

Tags

<html>, *<head>*, *<body>*, ... are **tags**

- they say what sort of thing is being created
- meaning varies hugely
  - *<head>* = metadata about page
  - *<title>* = name of page in browser
  - *<body>* = main contents of the page
  - *<h1>* = header (large text)
- should be opened *<head>* and closed *</head>* afterwards
  - not all tags need to be closed
- can contain other tags

Try it out!

- Create folder `elt_cs`
- Create file `index.html`
- Give it the following content:

```
<head>
        <title>My first file</title>
</head>
<body>
        <h1>Hello world</h1>
</body>
```

- Open it in a browser!

Useful tags

Tags define the structure of the page

- <*p*> = paragraph
- <*strong*> = text in **bold** type
- <*em*> = emphasised text (usually *italics)*
- <*ul*>, <*ol*> = unordered list, ordered list
- <*li*> = element in a list

We won't cover all possible tags here

- there are far too many!
- you should Google for more as you need them

Attributes

Sometimes, tags need extra information

<*img*> creates an image

- but what image?

<*img src='http://www.cs.st-andrews.ac.uk/images/prospectiveug/2.jpg'*>

- *src* is an **attribute** for the tag <*img*>
- it gives the 'source' of the image: its location, another URL or path

Many tags have important attributes

- <*a href="http://www.google.com"*> creates a link
- *href* is the target URL of the link

Optional attributes

Many tags have required attributes

Some attributes can be applied to **any** tag

- *<p title=”hello”>*
    - title: display “hello” when user passes mouse over paragraph
- *<ul id=’first_list’>*
    - id: allows you to refer to this element elsewhere (see later)
- *<li class=’in_first_list’>*
    - class: allows you to refer to a group of elements elsewhere

Structure & flexibility

All web pages have some basic features

- <*head*>, at start, contains metadata about page
- <*body*> afterwards contains concrete structure

- Technically, these can both be omitted
  - Web browser will fill in what it expects
  - This works for many, many tags/situations
- But don’t do this!
  - Browsers may be inconsistent with each other
  - Especially old browsers
  - Being explicit = consistent output across browsers

Readability

<!-- comments --> will not be shown to the user

- Used for making your code readable
- Helps you develop it

HTML doesn’t care about whitespace

- space, tab, new line are shown as a single space
- *<br>* tag inserts a real new line
- indentation “just” helps you read the HTML

```
<ul>
	<li>List item</li>
</ul>
```

A more complex example

- Update `index.html` with the following:
    - Feel free to add or change bits!

```
<head>
    <!-- Give this page a name -->
    <title>My second file</title>
</head>
<body>
    <h1>Hello world</h1>
    <p>My name is [your name here]. I attend the <a
        href=”http://www.st-andrews.ac.uk”>University of St Andrews</a>
</body>
```

File paths

We have seen a link to an external website

- *<a href="http://www.st-andrews.ac.uk">University of St Andrews</a>*

But what if we want to link to our own site?

Relative paths indicate files close to the current one

- slash (/) **within** path indicates a directory
    - *<a href="my_folder/my_file.html">File in subdirectory</a>*
- two dots (..) indicate the **parent** (container) directory
    - e.g. *<a href="../index.html">Up one level</a>*
- slash (/) at **start** indicates web root:
    - *<img src="/images/logo.png">*

Introducing CSS

HTML describes the **structure** of a page

- Not how that structure should be presented!
- What colours? What layout?

CSS governs the **style** of the page

- a new language!
- a new file

Basic CSS

- CSS gives **rules** for how to display things
        - Simplest “things” are tags

```
h1 {
        color:blue;
}
```

- These rules are collected in a file
        - e.g. *style.css*
- That file is included in the HTML
    - inside the <*head*> tag
    - *<link rel='stylesheet' type='text/css' href='style.css'>*

A few CSS possibilities

With CSS you can do many things:

- Make text bold, italic or underlined
  - *font-weight:bold;*
  - *text-decoration:underline;*
  - *font-style:italic;*
- Change colours
  - *color:red;*
  - *background-color:blue;*
- Indicate sizes
  - *height:150px;*
  - *width:50%;*

Colour in CSS

Computers output color through many **pixels**

- each is a combination of **red**, **green** and **blue** lights
- you can define how much of each is shown

You can define straightforward colours by name

- *color:red;*
- *colour:lightblue;*

You can specify components with numbers from 0 to 255

- *color:rgb(255,0,0);*

Or use hexadecimal numbers

- *color:#FF0000;*

Alternative CSS

- Can be included in main page

```
<head>
        <style type=’text/css’>
                h1 {
                        color:blue;
                }
        </style>
</head>
```

- Can even be included in tags
    - *<h1 style=’font-weight:bold;’>Text</h1>*

- Neither of these is recommended!