COMP2203: Application Scripting

Session DA2

Getting the Basics Right

David Argles

Electronics & Computer Science
University of Southampton

Haven Consulting
david@argles.org

Session DA2 Outline:

- Brief recap on HTML
- HTML versions
- Page naming conventions
- Requesting a web page
- Server- and client- side scripting
- Stateless design
- Designing for the web

Haven
Consulting
david@argles.org

A Brief Recap on HTML

- Web pages are transferred via HTTP
  - “**H**yper**Text** **T**ransfer **P**rotocol”. It's the **text** bit that's important - what comes in to your browser is *just* text
    [note to me - now demo this]
- The text can be formatted using HTML
  - “**H**yper**T**ext **Markup** **L**anguage” to specify how the text should be presented.
    [note to me - now demo tags in HTML]

A Brief Recap on HTML

- Different browsers may display the same page slightly differently...
  - *[note to me - now demo this]*
- ...or very differently!
  - older versions of IE are the worst offenders
- Separation of content and layout
  - You specify the page content
  - You **cannot** specify the user's page size

Haven Consulting
david@argles.org

HTML Versions

HTML1 to 4:

- Used capitals (<BODY>, <BR>)
- Was a bit sloppy about the markup (used <P> to start a paragraph, but nothing to note the end)

XHTML (strict):

- Uses lower case (<body>, <br />)
- Must have a closing tag (<p> closes with </p>, deprecated tags as in <br />)

HTML5

- Has given up the fight to some extent, but you should keep with the strict XHTML practices

Haven
Consulting
david@argles.org

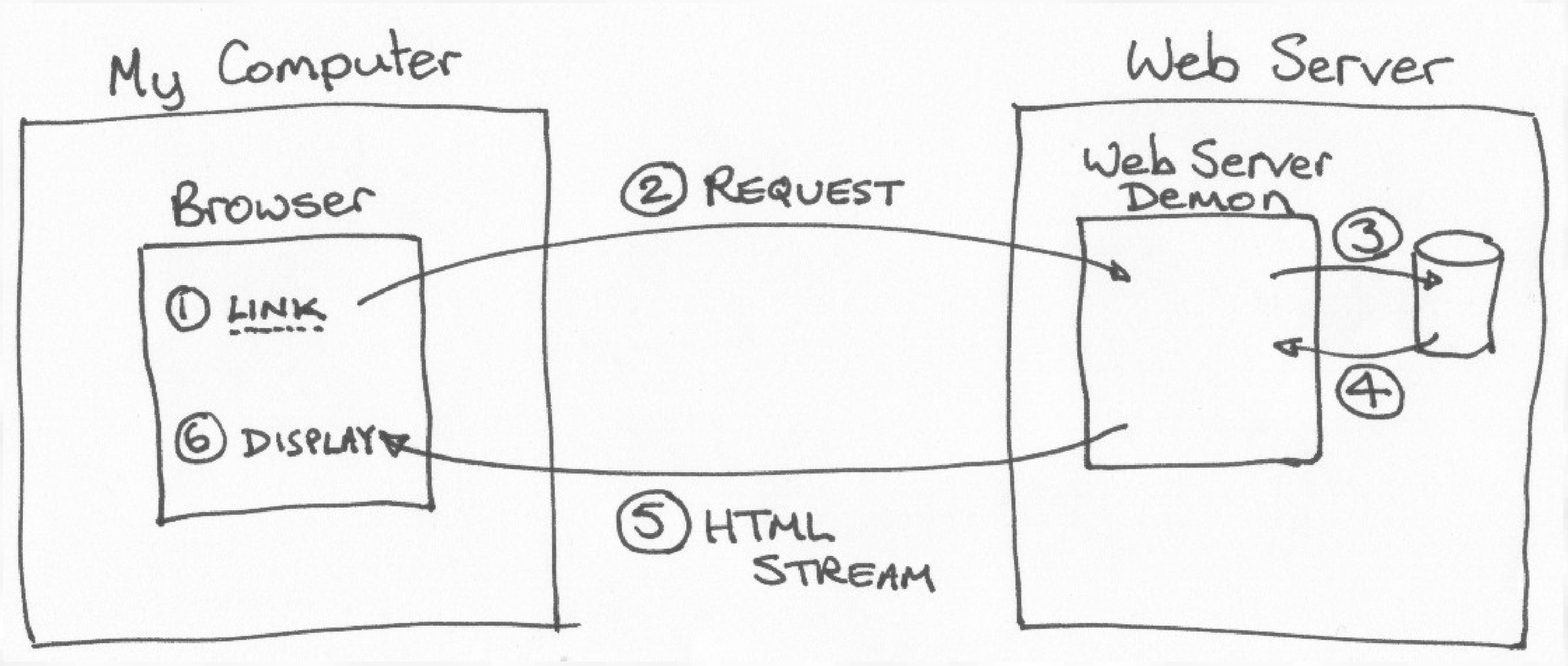
Page-naming Conventions

Use good page-naming conventions:

- Ordinary web pages should have an ".html" extension
- Do **not** use .htm – that was an MS corruption
- PHP pages will have to be called "xxx.php"
- If it's meant to be a web page, don't label it "xxx.txt" or whatever, call it what it really is
- Hence the need for proper tools like ConText; Wordpad will try and convert your file to rtf, and Notepad will try and add a .txt extension, so you'll end up with index.html.txt
- Make your main page "index.xxxx". This is a security feature, since it helps to prevent folk from snooping around where they shouldn't be

Requesting a Web Page

Now we need to think about how a web server works:



Requesting a Web Page

So, the routine is:

1 You click a link in your browser

2 Your browser makes a (text-based) HTTP request to the relevant server

3 The remote server assembles a stream of HTML. This *might* just be a case of getting an HTML file...

4 ...from the server's hard disc, but it *might* require the server to create a page on the fly, using a linked database, for example

5 The server returns the HTML stream to your machine

6 Your browser displays that HTML as a web page

Requesting a Web Page

- The user doesn't see all that is going on
  - *[note to me - show http://argles.org/david/ and http://argles.org/oop/]*
- Even with an identical HTML stream the server code might be completely different

Server- v. Client- Side Scripting

The implications of this are:

- Writing a page of HTML, saving it on your own computer, and dragging it into your browser is **not** the same thing as storing that page on a server and viewing it over the Internet
- There are *two* places where code can run to modify what happens when you call up a web page; on your computer, or on the server. If you use code that runs on your computer, it is called *client-side* scripting; if it runs on the server, it is *server-side* scripting. These two do very different jobs, so <script> code (client-side) and php code (server-side) are not two ways to do the same thing, although it may sometimes look like it

Stateless Design

In this process, HTTP is designed to be "stateless":

- The basic design of the web is that random clients make unconnected page requests of the servers
- So the server is not expected to remember anything about previous requests
- This is a bit of a problem when dealing with on-line transactions, e.g. Internet purchases
- So there have to be fixes. But we'll deal with that later

Designing for the Web

Then there's the issue of design:

- We tend to have a model of paper publishing in our heads when approaching the Internet
- In paper publishing, we know the paper size, quality, range of inks, etc, etc
- In web publishing, we have a stream of text with tags to identify what might be special about the text. We have ***no idea*** what shape/size/colour range etc is available on the client computer
- So we ***must*** get into the habit of realising that we are preparing page content, specifying preferences for presentation, and ***leaving it to the client to render the page appropriately***

Haven
Consulting
david@argles.org

Break

Developing web stuff via a tablet

Developing web stuff ***for*** a tablet. Or a phone, or a “phablet”, or...