Information Management Resource Kit

Module on Management of Electronic Documents

UNIT 6. NETWORKING DOCUMENTS AND DATABASES

LESSON 1. WEB STANDARDS AND STATIC WEBSITES

NOTE

Please note that this PDF version does not have the interactive features offered through the IMARK courseware such as exercises with feedback, pop-ups, animations etc.

We recommend that you take the lesson using the interactive courseware environment, and use the PDF version for printing the lesson and to use as a reference after you have completed the course.

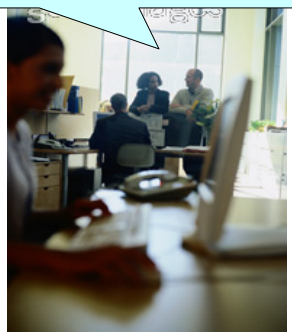
Objectives

At the end of this lesson, you will:

- have acquired knowledge of the communication **standards** on the Web, and
- be able to understand the capabilities and limitations of a **simple static website**.

Introduction

We will build a small document management website in order to disseminate our documents with simple bibliographic information.

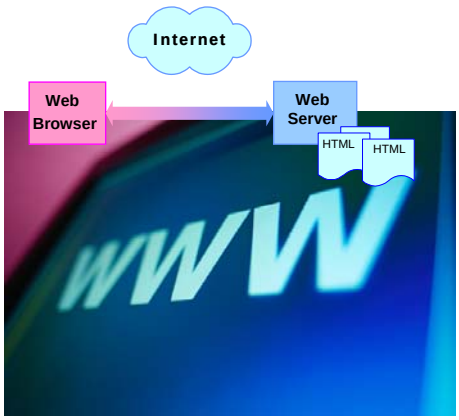


How can you make resources available to users on the Web?

The simplest way is to build a simple **website**.

Before looking at how to do it, let's start with some basic concepts to understand how you communicate with your users on the Web.

Uniform Resource Identifier



The World Wide Web, or simply Web, is a way of accessing information over the Internet.

Together with e-mail, the Web is the most popular way to disseminate information over the Internet.

The Web is a **client/server** application:

- web-servers are computers that deliver (serve up) web documents called web pages;
- to access a web page, users connect to the web-server using a web browser (client), such as Internet Explorer or Netscape.

Uniform Resource Identifier

On the Web, resources are identified by a scheme called **URI (Uniform Resource Identifiers)**. E.g.: **http://www.fao.org**.

A **URI** may include four parts:

http://	**www.fao.org**	**/path/ask.cgi**	**?x=1&y=2&z=3**

Click on each part to view the description

SCHEME

Identifies the way in which the other parts of the URI syntax are to be used.

The most common form of a URI is a **Uniform Resource Locator (URL)** which identifies a resource from its network location.

Uniform Resource Identifier

www.fao.org

AUTHORITY

A top hierarchical element for a naming authority, which for the http scheme is the Internet domain name.

/path/ask.cgi

PATH

Data, specific to the authority, identifying the resource within the scope of that scheme and authority (e.g. the file name in the http scheme).

?x=1&y=2&z=3

QUERY

A string of information to be interpreted by the resource.

Uniform Resource Identifier

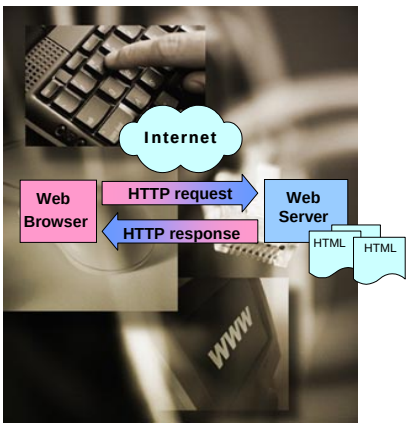
For example, can you identify the parts of the following URI?

http://www.w3c.it/talks/XMLDays2002/overview.htm

	SCHEME	AUTHORITY	PATH	QUERY
talks/XMLDays2002/overview.htm	○	○	○	○
http://	○	○	○	○
www.w3c.it	○	○	○	○

Click on your answers

HTTP Protocol



The Web uses the HTTP protocol to transmit data.

HTTP is a simple **request/response** protocol: the client sends an HTTP request message and the server sends back an HTTP response.

For example, if you enter the URL *http://www.fao.org/index.html* in your browser, this sends a request to the server whose domain name is *fao.org*. The server then fetches the page named *index.html* and sends it to your browser.

HTTP is also a **stateless** protocol: the server can handle a series of messages from a client, but it responds to each one individually, **one at a time**, without making any connection between them.

HTTP Protocol

PRINT TABLE

An HTTP message must have a precise **structure**.

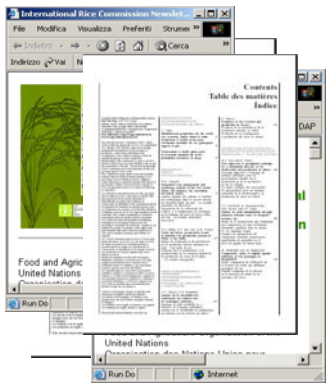
Here is a typical HTTP request to retrieve the file index.html, with the relevant response:

HTTP REQUEST MESSAGE[1]	DESCRIPTION
GET /path/index.html HTTP/1.1	Request line. It contains method (here GET[2]), name of target resource, HTTP version.
Accept: text/html User-agent: IE5	Header Fields (0 o +): sequence of name:value pairs.
HTTP RESPONSE MESSAGE	**DESCRIPTION**
HTTP/1.1 200 OK	Response header
Mime_version: 1.0 Content_type: text/html Content_length: 2000	Response header fields
<HTML> <HEAD> <TITLE> … </TITLE> </HEAD> <BODY> ….. </BODY> </HTML>	Entity body (here it is the content of the file index.html)

[1] A request can also optionally contain an entity body.

[2] GET is one of the 13 methods in HTTP/1.1, which support various manipulations of resources on the server from the remote client.

Building a Simple Static Website



So, a website is a site (location) on the World Wide Web.

Each website contains a home page, which is the first document users see when they enter the site.

A site can also contain additional documents and files. Let's imagine you want to deliver a set of documents and show simple bibliographic information for each one, e.g.:

- Document title
- Publication date
- Author
- Language
- Format
- Description

You will build a **simple static website**.

Building a Simple Static Website

First, you have to design the **site map**.

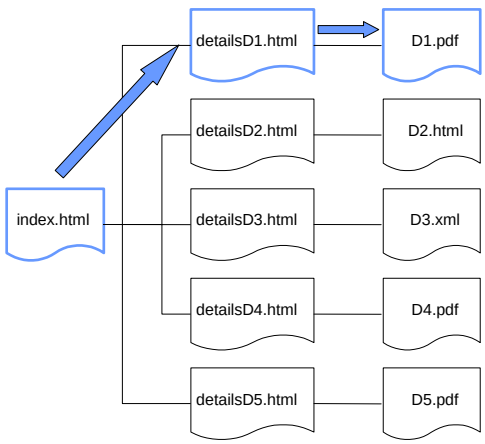
The graph on the right shows an example of site map for five documents, in several formats.

The first page (index.html) will display the **list of documents**.

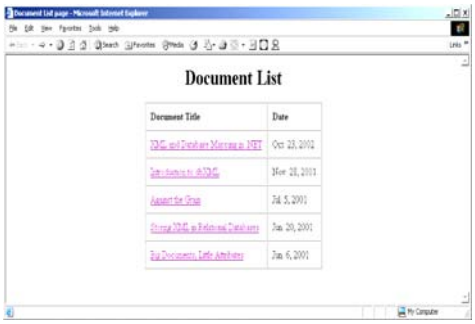
When selecting one from the list, the **document details** for the selected document appear.

From that second page, you can view the **document itself**.

Now, let's look at what the result would be...



Building a Simple Static Website



The first page would be similar to the one on the left.

To access the first document, the user would have to:

- select the **document title**, then
- select the format, "**PDF**", to bring up the document itself.

View Animation

Building a Simple Static Website

The first step to achieve your goal is to build the "Document List" page (index.html).

This is a fragment of how the HTML page should look:

```
…
<h1 align="center">Document List</h1>
<table cellpadding="10" cellspacing="0" border="1" align="center">
  <tr>
    <th align="left">Document Title</th>
    <th align="left">Date</th>
  </tr>
  <tr>
    <td><a href="detailsD1.html">XML and Database Mapping in .NET</a></td>
…
```

The **<a href="detailsD1.html">** specifies a hypertext link to another HTML document.

The "detailsD1.html" page contains the document information for the "XML and Database Mapping in .NET" document.

Each document has its own "detailsD#.html" page.

View the entire HTML page

Building a Simple Static Website

Now, you have to build the “detailsD1.html” page.

This is a fragment of how this page should look:

```
...
<tr>
   <th align="left">Language</th>
   <td>EN</td>
  </tr>
  <tr>
   <th align="left">Format</th>
                <td><a href="D1.pdf">PDF</a></td>
  </tr>
...
```

On the “detailsD#.html” pages there is a **hypertext link to the PDF**.

Selecting “PDF” on the page will bring up the PDF in the browser.

View the entire HTML page

Building a Simple Static Website

Imagine you want to add a sixth document, entitled: “Introduction to HTML”, to the list.

Where would you have to enter the following codes?

detailsD6.html

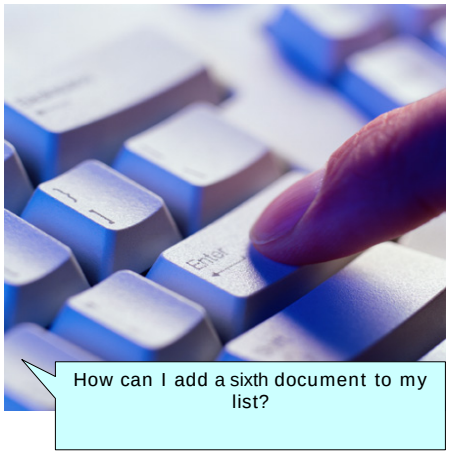
```
<td><a href="D6.pdf">PDF</a></td>
```

index.html

```
 <td><a href="detailsD6.html">
Introduction to HTML </a></td>
```

Click on each option and drag it to the relevant box.

Limitations of Simple Static Website



To **add a new document** to the list, you would have to:

- **create a new "detailsD#.html" page** with the document details (e.g. "detailsD6.html"), and
- **modify the index.html (Document List Page)** by adding the new document's title, date and hypertext link to its detailsD#.html page.

This means that you must edit the index.html whenever you add a new document.

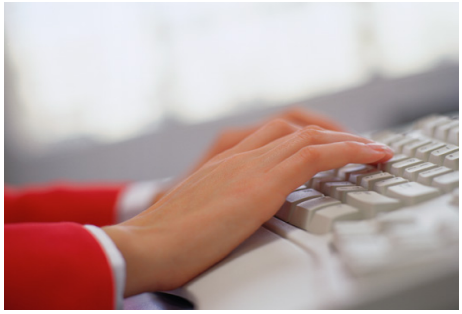
This is only one of the limitations of a simple static website…

Limitations of Simple Static Website

A simple static website has other limitations as a client/server application:

- **Information is repeated** in the home page (index.html) and each of the details pages, so you need to be careful when we update the pages to make sure the information is consistent (not to mention the extra effort involved in typing out the same information twice).
- **We can't sort the list** of documents, since it's a static list, fixed in index.html.
- **What happens if the list becomes very long?** You can't have a sequence of pages with a limited number of documents listed on each one.
- **Users can't search** for documents.

Limitations of Simple Static website



There's another general limitation of static websites, which comes about because we don't have access to programming logic: we can't have a user 'session' which remembers what the user has done so far and adjusts the behaviour of the applications as a result.

Limitations of Simple Static website

Some of the limitations of static websites have been overcome by **dynamic websites**.

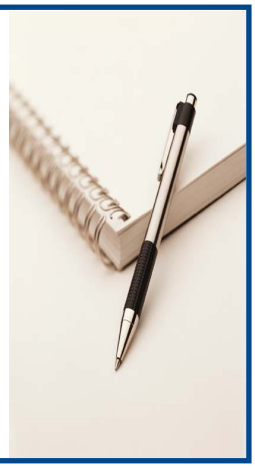
A dynamic website allows users to interact with program logic on the web-server.

The information or content for the web pages is stored in the database and is revealed when requested through a web browser by the user.

There are many technologies for producing dynamic sites, including **CGI** (Common Gateway Interface), **Java** technologies (servlets and Java Server Pages), **ASP** (Application Server Pages), **PHP** (Personal Home Page).

Summary

- **TCP / IP** is the most popular of all networking standards: it identifies the devices on a network and ensures that the data are correctly transmitted.
- If we want to link more networks together, then we can use a **router**, a device which knows about the IP numbers on each network.
- Resources are identified on the Web by a scheme called **URI (Uniform Resource Identifier)**.
- In a simple static website, **information is repeated** both in the home page (index.html) and each of the details pages, so we need to be careful when we update the pages.
- Moreover, we can't have a user 'session' which **remembers what the user has done** so far and adjusts the behaviour of the applications as a result.



Exercises

The following three exercises will allow you to test your understanding of the concepts described until now.

Good luck!

Exercise 1

Can you associate the following elements to their definitions?

TCP	A number that identifies the devices connected to the network.
Router	A protocol that ensures the correctness of the information transmission.
IP number	A device that knows the IP numbers on different connected networks.

Click on each option and drag it to the relevant box.

Exercise 2

In which kind of HTTP message are included the following parts?

```
<HTML>
 <HEAD> <TITLE> ... </TITLE> </HEAD>
<BODY> ..... </BODY>
</HTML>
```

```
GET /path/index.html    HTTP/1.1
```

HTTP request

HTTP response

Click each option, drag it and drop it in the corresponding box.

When you have finished, click on the confirm button.

Exercise 3

Which of the following is a limitation of a simple static website?

- It can contain only a limited number of resources.
- It can't contain multimedia resources.
- Updating its content can take a lot of time.

Click on your answer

If you want to know more...

The technology of the Internet has been developed by the Internet Engineering Task Force (IETF) which provide a useful starting point for research (www.ietf.org)

From its early days, the Internet has been 'managed' by the Internet Society (www.isoc.org)

The World Wide Web Consortium sets information standards for the Web (W3C – www.w3.org)

The official source of information on how to address resources on the Web (www.w3.org/Addressing)

Open source web-servers are available from the W3C (Jigsaw - www.w3.org/Jigsaw) and the Apache Software Foundation (www.apache.org).

Spainhour, S. & Eckstein, R. 2003. Webmaster in a Nutshell. O'Reilly UK; ISBN: 0596003579