

# Introduction to Server-Side Web Languages

## Server-Side Web Languages

Uta Priss  
School of Computing  
Napier University, Edinburgh, UK

# Outline

Learning Outcomes and Assessments

Server-Side Web Languages

# Learning Outcomes

- By the end of this module, students will have an understanding of
- ▶ Server-side web-based programming and the CGI environment.
  - ▶ Web-based scripting languages, their advantages and problems.
  - ▶ Searching and pattern matching using regular expressions.
  - ▶ Implementation of a small web-based server-side application.
  - ▶ Some current technical and research issues in this field.

# Assessments

This module has two assessments: an exam and a coursework.  
Each counts 50%.

# Learning Computer-related Subjects

- ▶ Changing technologies require “life-long learning”.

# Learning Computer-related Subjects

- ▶ Changing technologies require “life-long learning”.
- ▶ Knowledge about computers is never complete.

# Learning Computer-related Subjects

- ▶ Changing technologies require “life-long learning” .
- ▶ Knowledge about computers is never complete.
- ▶ It is usually better to know where and how to find solutions to arising problems than to memorise solutions that worked in the past.

# Learning Computer-related Subjects

- ▶ Changing technologies require “life-long learning” .
- ▶ Knowledge about computers is never complete.
- ▶ It is usually better to know where and how to find solutions to arising problems than to memorise solutions that worked in the past.
- ▶ It is useful to maintain logbooks (weblogs) to avoid having to search for the same solution twice.



# Learning Computer-related Subjects

- ▶ Changing technologies require “life-long learning” .
- ▶ Knowledge about computers is never complete.
- ▶ It is usually better to know where and how to find solutions to arising problems than to memorise solutions that worked in the past.
- ▶ It is useful to maintain logbooks (weblogs) to avoid having to search for the same solution twice.
- ▶ The best computer textbook is

# Learning Computer-related Subjects

- ▶ Changing technologies require “life-long learning” .
- ▶ Knowledge about computers is never complete.
- ▶ It is usually better to know where and how to find solutions to arising problems than to memorise solutions that worked in the past.
- ▶ It is useful to maintain logbooks (weblogs) to avoid having to search for the same solution twice.
- ▶ The best computer textbook is Google (or similar search engines).

## Module Structure: Tutorials

- ▶ The learning materials are arranged in exercises simulating real web applications.
- ▶ The learning materials are more technology-oriented than fact-oriented. Students are not required to learn anything “by heart”.
- ▶ All assessments are “open-book” and simulate problems as occurring in real web applications.
- ▶ On-line materials (including search engines) should be used for the coursework - but all resources must be quoted to avoid plagiarism.

## Module Structure: Lectures

- ▶ The lectures support the tutorials.
- ▶ The lectures include some structured presentations (for introduction and overview of topics) but also time for class discussion and opportunities for asking questions.
- ▶ Due to the nature of the subject matter, the lectures will not be “complete”.
- ▶ Students are encouraged to complement the lecture materials by reading web resources. The module website provides some starting points.

The reason for providing learning materials in a more flexible manner is to encourage students to take ownership and control of their learning.

## Getting Help with the Module Materials or Coursework

- ▶ Ask questions during the tutorials or lectures. (Check the timetable for additional tutorial sessions.)
- ▶ Email questions to the lecturer.
- ▶ Check the module website for FAQs and announcements.
- ▶ Type your question into Google. (Note: use quotation marks around phrases, eg “Server-side web languages”).
- ▶ Ask fellow students (but not if the problems relate to the coursework!)
- ▶ Request an appointment with the lecturer.

# Server-Side Languages

Server-side languages are implemented and executed in a webserver environment.

- ▶ Typical server-side languages are Perl, Php, Python, Asp.
- ▶ A typical client-side language is Javascript.

Java can be used either server-side (as Java Servlets) or client-side (using applets).

# Static HTML pages

Many HTML documents provide **static** content, which is

- ▶ stored on a webserver,
- ▶ retrieved via the HTTP protocol,
- ▶ displayed to a client via a browser.

# Dynamic Content

Some HTML documents provide **dynamic** content. This content

- ▶ is generated by a computer program;
- ▶ can retrieve information from a database;
- ▶ can respond to a specific user request (e.g. a webform);
- ▶ is converted into an HTML page;
- ▶ which is retrieved via HTTP by the user's browser.



# Advantages of Dynamic Content

## Dynamic content

- ▶ is more flexible than static content (e.g. on-line newspapers);
- ▶ can respond to specific user requests (e.g. e-commerce);
- ▶ can collect user information (e.g. on-line surveys, guestbooks);
- ▶ can provide an interface to a database (e.g. search engines);
- ▶ can facilitate basic user interaction (e.g. on-line shopping).

## Server-side or Client-side?

Which of these are better implemented client-side or server-side?

- ▶ Applications with many graphics (e.g. certain computer games) -

## Server-side or Client-side?

Which of these are better implemented client-side or server-side?

- ▶ Applications with many graphics (e.g. certain computer games) - Client-side (because graphics are slow over the web)

## Server-side or Client-side?

Which of these are better implemented client-side or server-side?

- ▶ Applications with many graphics (e.g. certain computer games) - Client-side (because graphics are slow over the web)
- ▶ Database interfaces -

## Server-side or Client-side?

Which of these are better implemented client-side or server-side?

- ▶ Applications with many graphics (e.g. certain computer games) - Client-side (because graphics are slow over the web)
- ▶ Database interfaces - Server-side (because the user usually only needs a few records from a database; transmitting the whole database would be slow)

## Server-side or Client-side?

Which of these are better implemented client-side or server-side?

- ▶ Applications with many graphics (e.g. certain computer games) - Client-side (because graphics are slow over the web)
- ▶ Database interfaces - Server-side (because the user usually only needs a few records from a database; transmitting the whole database would be slow)
- ▶ On-line maps -

## Server-side or Client-side?

Which of these are better implemented client-side or server-side?

- ▶ Applications with many graphics (e.g. certain computer games) - Client-side (because graphics are slow over the web)
- ▶ Database interfaces - Server-side (because the user usually only needs a few records from a database; transmitting the whole database would be slow)
- ▶ On-line maps - Server-side (because a user who searches for an address does not need to download a whole atlas).

But many applications (such as on-line banking applications) have both client-side components (using Javascript or Java for better usability and graphics of the client browser window) and server-side components for storing the data in a database on the server.

# Challenges for Server-Side Applications

The biggest challenge is **Security!** The user can never be trusted because for each click the user makes on a browser a new connection is established.



# Challenges for Server-Side Applications

The biggest challenge is **Security!** The user can never be trusted because for each click the user makes on a browser a new connection is established.

A second challenge is the limitation of the HTTP protocol. User activities are limited by what is possible via HTML and common browsers.

## Using a Non-Napier Webserver

As a learning experience, students may want to explore using a webserver outside the university.

It is nowadays possible to obtain free webspace (if one accepts ad banners) or for about £20-40 per year without ads. The webserver used needs to support Perl/CGI and PHP. (Domain names cost from about £7 for 2 years).

Please, note that webspace and domain names in other countries may be governed by different laws. Napier University will not be responsible for any problems (technical, legal, financial or other) that students encounter if using a non-Napier resource.