



Web Application Development 2

Course admin

- Lecturer: Dr. Alistair Morrison
 - email: alistair.morrison@glasgow.ac.uk
- Lectures: All made available as videos on Moodle at the start of each week. Live interactive session **Fri 1100-1200**.
 - Live sessions will be opportunity for Q&A
 - Attendance encouraged but non-compulsory to live lecture sessions
- Lab sessions: You **are** expected to attend labs every week. Details TBC about in-person/online etc – will be announced by wk 2.
- Moodle page:
<https://moodle.gla.ac.uk/course/view.php?id=29970>

What is a Web App?

- It is a Distributed Information Management system (DIM)
- Enables the management, sharing, finding, modification and presentation of information
- Does so over a network, in a distributed fashion
- Typically has many users, often geographically separated
- Ideally DIMs enable users to access timely, relevant and useful information in a seamless manner
- Think MyCampus ☺

Examples



[Booking.com](#)



[YouTube](#)



[the trainline](#)

your first stop for train tickets

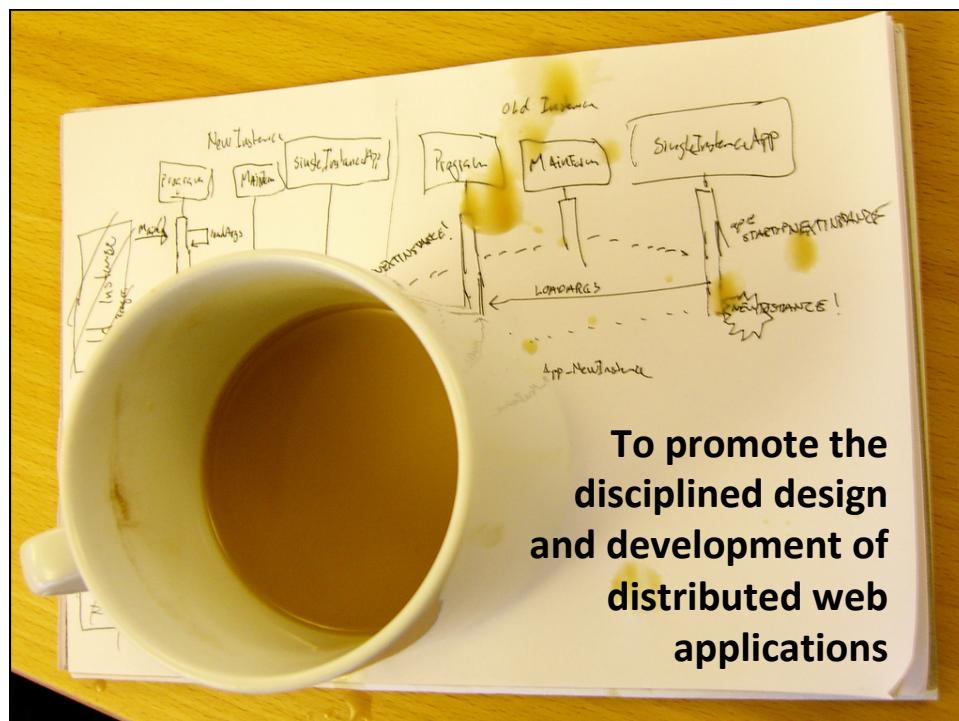
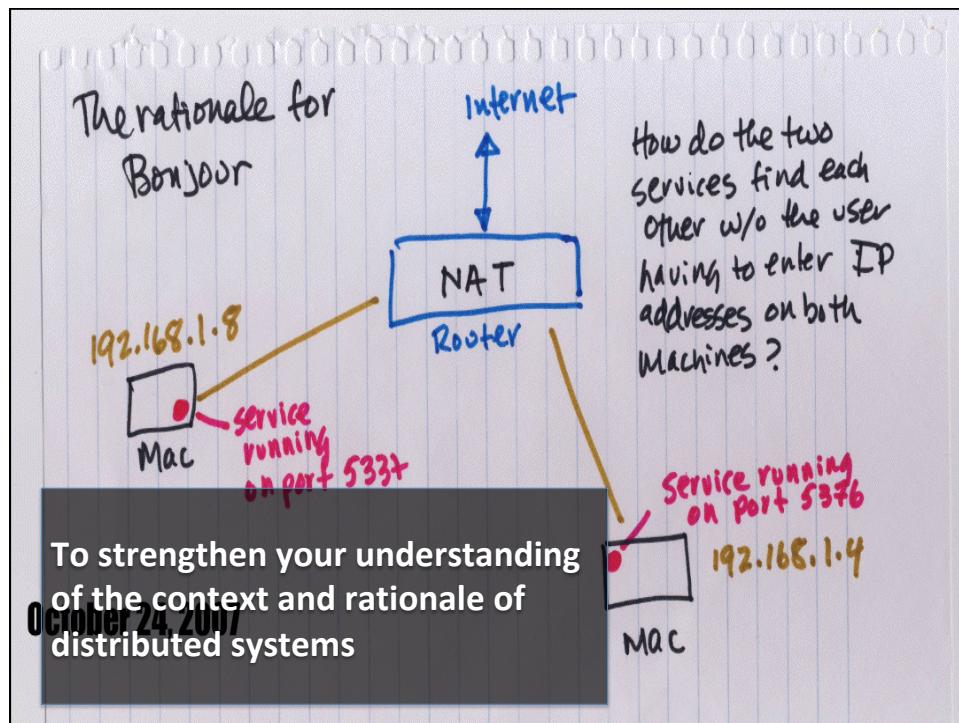
[amazon](#)

[lastminute.com](#)

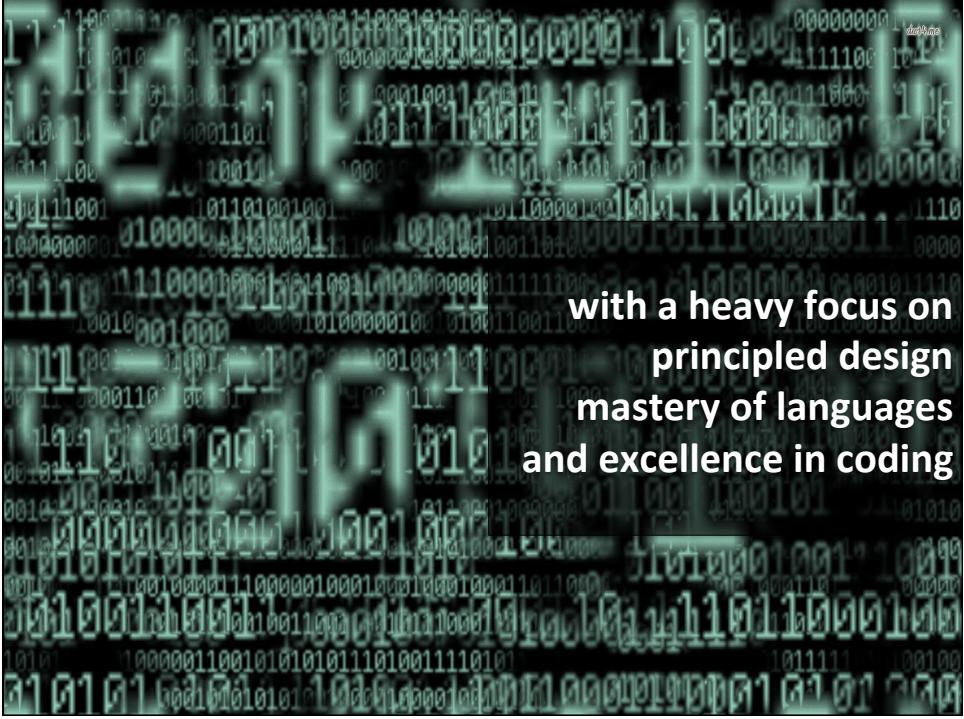
COURSE AIMS

To provide an overview of
the tools and technologies
used in web development









**with a heavy focus on
principled design
mastery of languages
and excellence in coding**

Types of Architecture

System Architecture

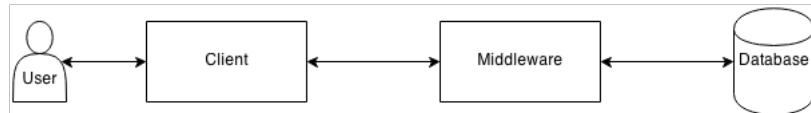
Information Architecture

Design Elements

- System Architecture (system focused)
 - Specifications and Requirements
 - High Level System Architecture
 - Entity Relationship Diagrams
 - Sequence Diagrams
- Information Architecture (user focused)
 - User Personas
 - User needs matrix
 - Site design / URL design
 - Wireframes and Walkthroughs

SYSTEM COMPONENTS

High Level System Architecture



User and Client

- The **User** could be human or machine, which
 - Initiates contact and interacts with the client
 - Ranges in skills and abilities
 - Has a number of requirements that need to be satisfied
- The **Client** is a program sitting on a client device which
 - sends **request messages**
 - accepts **response messages**
 - acts on the message
 - either communicating to the user
 - or affecting the environment in some way

Messaging

- The **request message** is sent from the client to the server to:
 - ask for some information
 - send some information to be stored
 - either from user input
 - or from some device (sensor)
- The **response message** is sent from the server to a client to:
 - return the requested information
 - effect some change in the environment

Messaging Protocols

- The **request message protocol**
 - is usually an HTTP request
 - embedding any data to be sent
- The **response message protocol**
 - is an HTTP response
 - with content that is often in JSON/XML

Middleware & Backends

- The **Middleware/Application Server** is the central component which
 - accepts **request messages** from clients
 - returns **response messages**
 - co-ordinates the application components
- The **Backend/Database** is typically on a separate node and
 - stores the data for the application
 - provides the data when needed
 - needs to be scalable and reliable
 - could be a database, an index, a flat file



Web Development Complexity

- **Collision of Languages**
 - Markup Languages
 - Programming Languages
 - Database Query Languages
- **Shifting Standards**
 - Document Object Model
 - XML/JSON
- **Web Browser Compatibility**
 - Browser wars encourage new ‘unique’ features
- **HTTP is a Stateless Protocol**
 - But most applications require the persistence of state

Signs of Hope

- **Web development has become a serious business**
 - Despite all these challenges, there are many incentives to persevere
- **In line with this, the methods of development are maturing and increasingly adopting good practices of ‘classical’ Software Engineering:**
 - Application Programmer Interfaces (APIs)
 - Libraries
 - Frameworks
 - Tools
 - Standards

Tools



Web Development Tools

- **The nature of web development is disjoint**
 - a developer must become familiar with a set of distinct and (typically) non-integrated tools
- **Web development tool support is not yet as advanced as with ‘classic’ software development**
 - Most languages have several complex IDEs (Integrated Development Environments) to choose from
 - IDLE, Eclipse, PyCharm, etc

What you code in...

- An important tool is the text editor or IDE
- The choice of text editor and your expertise in its usage can seriously affect your productivity
 - Syntax Awareness
 - Auto-completion
 - Snippets
 - Scripts & Macros
 - Integration with other development tools (revision control)
- Some IDEs have plugins for scripting languages (e.g., Eclipse has a PHP plug-in)

Checking your code

- Interpreted languages lack the compilation stage where errors and warnings can be raised
- Frustratingly, scripts will just run until they reach an error and fall over
 - If you are lucky, there will be an error message
 - But it won't make much sense most of the time
- There are a wide range of tools that will perform static analysis of scripts to spot errors
 - PyLint / JSLint
 - PHP_CodeSniffer (works on PHP, Javascript & CSS)
 - Developer Tools in Chrome, Firefox, Safari...

Understanding W3C standards

- Each aspect of the software is specified by a W3C (<https://www.w3.org/>) **activity** run by a working group which produces the following reports:
 - **working draft** - no consensus yet
 - **candidate recommendation** - published in order to gather implementation experience and feedback
 - **proposed recommendation** - sent for final approval by advisory committee
 - **w3c recommendation** - approved by the W3C
- Note that this means everything evolves (and gets out of date) very fast

Complying with Standards

- There are a variety of tools to check that certain parts of a web application conforms to standards
 - (X)HTML
 - <http://validator.w3.org/>
 - CSS
 - <http://jigsaw.w3.org/css-validator/>
 - XML Feeds (RSS/Atom)
 - <http://validator.w3.org/feed/>
 - JavaScript Code Quality Tool
 - <http://www.jslint.com/>



Main Technologies in this Course

- Python
- Django
- HTML
- CSS
- HTTP GET/POST
- XML/JSON
- XHTML
- JavaScript
- JQuery
- AJAX
- As well as:
 - GitHub
 - Pip
 - PythonAnywhere
 - IDLE
 - Virtual Environments
 - Draw.io / Diagrams.net

Lots to Learn

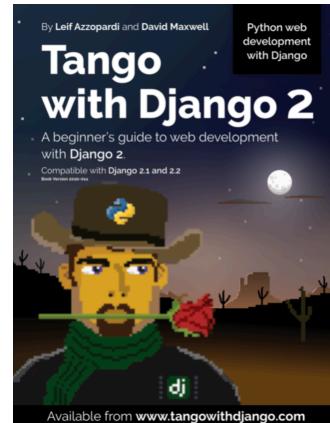
- Developing Web Applications requires the use of many different languages, technologies, protocols, standards, and formats.
- You will be expected to be proficient in each of these technologies
- You will need to understand how the different components fit together
- You will need to answer code-based questions in the exam

Assessments

	<u>Deadlines</u>
• (1) Lab Exercises (10%) – development of the Rango application	11 Feb
• (2) Group Project (40%) – Design specification (10%) – Project presentation (5%) – Project application (25%)	25 Feb 25 Mar 25 Mar
• (3) Moodle quizzes (10%) – Multiple-choice questions	Every Friday (starting 2nd week)
• (4) Exam (40%)	

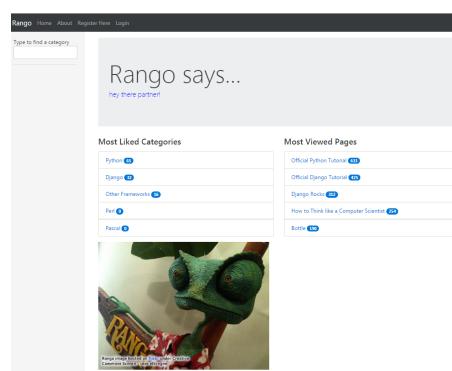
Rango application

- Main focus of practical work/labs in weeks 1-5
- Developed in Python and Django using “Tango with Django” (updated for v2.1 & 2.2), by Leif Azzopardi and David Maxwell
 - Version **2020-01a**
- Obtain the book via course Moodle page



Rango application (cont)

- To be developed individually
- Will be marked using automated tests
- Deadline 11 Feb



WAD2 Project

- In small groups, you will be responsible for the design, development and deployment of a web application using Python / Django
- This will be the main focus of lab sessions 6-11
- Choice of what to build is up to you
 - but it will have to be designed well!
 - design specification assesses this
- You will present your application to your lab group on 24 / 25 March
- And submit the code by 25 March

Moodle quizzes

- Quizzes will aim to reinforce concepts, promote interaction, feedback
- Typically four to five questions per week, starting next (2nd) week
- Released onto Moodle over the weekend
- Deadline will be Friday mornings 10.30am
 - Live lecture session at 11.00 will cover answers
- Designed to be relatively quick and straightforward, and based on recent lectures
- If ill, see the course handbook (available on Moodle)