

# COMP30680

## Web Application Development

Wrap up - Frameworks: an introduction

David Coyle  
[d.coyle@ucd.ie](mailto:d.coyle@ucd.ie)

# Learning outcomes

On successful completion of this module the learner will be able to:

1. Develop client-side applications using HTML, CSS and JavaScript.
2. Become familiar with the JSON data-interchange format and RESTful web service APIs.
3. Develop server-side applications using PHP.
4. Connect to a database from a web application.
5. Implement web applications using LAMP/WAMP/MAMP solution stacks.
6. Display an overall awareness of how to approach website development.
7. Gain experience of programming in different languages (quickly).

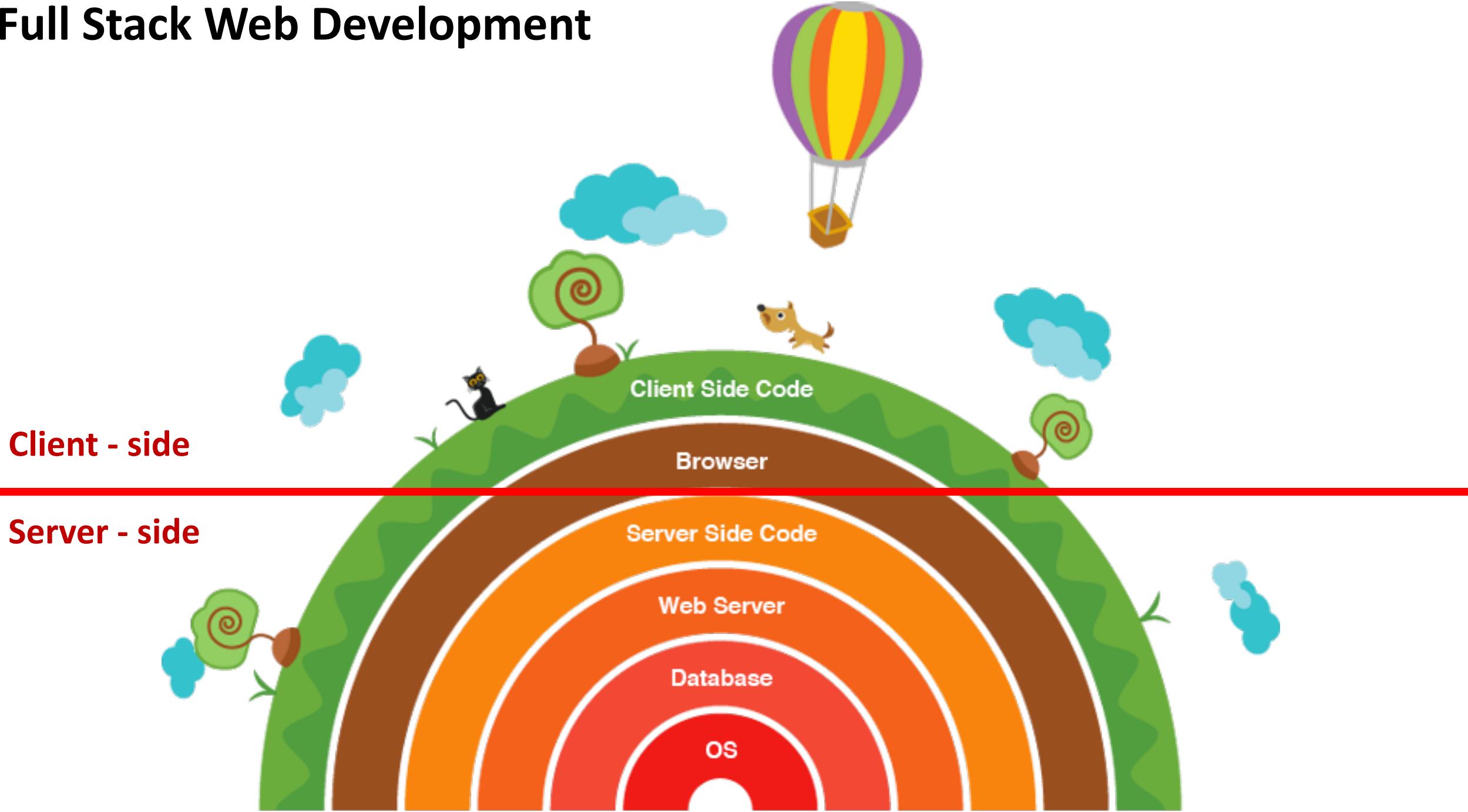
# Core skills



# Full Stack Web Development



# Full Stack Web Development



# Frameworks

A **web framework (WF)** or **web application framework (WAF)** is a software framework that is designed to support the development of web applications including web services, web resources and web APIs.

Web frameworks aim to *alleviate the overhead* associated with common activities performed in web development.

For example, many web frameworks provide *libraries for database access*, *templating frameworks* and *session management*, and they often promote code reuse.

Though they often target development of dynamic websites they are also applicable to static websites.



Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

[Download Bootstrap](#)

Currently v3.3.6

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs.

For examples of sites using Bootstrap see: <https://expo.getbootstrap.com/>



**jQuery** is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.



**Ajax** is a set of web development techniques using many web technologies on the client-side to create asynchronous Web applications. With Ajax, web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page.



HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. **AngularJS** lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.



# Data-Driven Documents



**D3.js** is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

See [more examples](#).

# JavaScript Frameworks

Project	Ajax	MVC framework	MVC push-pull	I18n & L10n?	ORM	Testing framework(s)	DB migration framework(s)	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)
AngularJS	XHR, JSONP ↗	Yes		I18n and L10n ↗		Karma (unit testing) ↗, Protractor (end-to-end testing) ↗		Content Security Policy (CSP) ↗, XSRF ↗	Templates ↗	Caching ↗	Form validation (front-end) ↗
EmberJS	Yes	Yes		Yes	Ember Data ↗	QUnit ↗			Handlebars ↗		
qooxdoo	Yes	Data binding ↗		I18n ↗		Testrunner ↗					Form Validation ↗
SproutCore	Yes	Yes									
Wakanda	Yes	Yes	Push & Pull		Native Object NoSQL DB ↗	CommonJS Unit Testing ↗ YUI Test Service ↗		Data Security and Access Control ↗		Storage (application.storage, user.storage, SessionStorage) ↗	

# JavaScript Frameworks

Project	Ajax	MVC framework	MVC push-pull	I18n & L10n?	ORM	Testing framework(s)	DB migration framework(s)	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)
AngularJS	XHR, JSONP ↗	Yes		I18n and L10n ↗		Karma (unit testing) ↗, Protractor (end-to-end testing) ↗		Content Security Policy (CSP) ↗, XSRF ↗	Templates ↗	Caching ↗	Form validation (front-end) ↗
EmberJS	Yes	Yes		Yes	Ember Data ↗	QUnit ↗			Handlebars ↗		
qooxdoo	Yes	Data binding ↗		I18n ↗		Testrunner ↗					Form Validation ↗
SproutCore	Yes	Yes									
Wakanda	Yes	Yes	Push & Pull		Native Object NoSQL DB ↗	CommonJS Unit Testing ↗ YUI Test Service ↗		Data Security and Access Control ↗		Storage (application.storage, user.storage, SessionStorage) ↗	

**AJAX** = Asynchronous JavaScript and XML.

## What You Should Already Know

Before you continue you should have a basic understanding of the following:

- HTML
- JavaScript

If you want to study these subjects first, find the tutorials on our [Home page](#).

---

## What is AJAX?

AJAX = Asynchronous JavaScript and XML.



AJAX is a misleading name. You don't have to understand XML to use AJAX.

AJAX is a technique for creating fast and dynamic web pages.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.

Examples of applications using AJAX: Google Maps, Gmail, YouTube, and Facebook.

# JavaScript Frameworks – client side

Project	Ajax	MVC framework	MVC push-pull	I18n & L10n?	ORM	Testing framework(s)	DB migration framework(s)	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)
AngularJS	XHR, JSONP ↗	Yes		I18n and L10n ↗		Karma (unit testing) ↗, Protractor (end-to-end testing) ↗		Content Security Policy (CSP) ↗, XSRF ↗	Templates ↗	Caching ↗	Form validation (front-end) ↗
EmberJS	Yes	Yes		Yes	Ember Data ↗	QUnit ↗			Handlebars ↗		
qooxdoo	Yes	Data binding ↗		I18n ↗		Testrunner ↗					Form Validation ↗
SproutCore	Yes	Yes									
Wakanda	Yes	Yes	Push & Pull		Native Object NoSQL DB ↗	CommonJS Unit Testing ↗ YUI Test Service ↗		Data Security and Access Control ↗		Storage (application.storage, user.storage, SessionStorage) ↗	

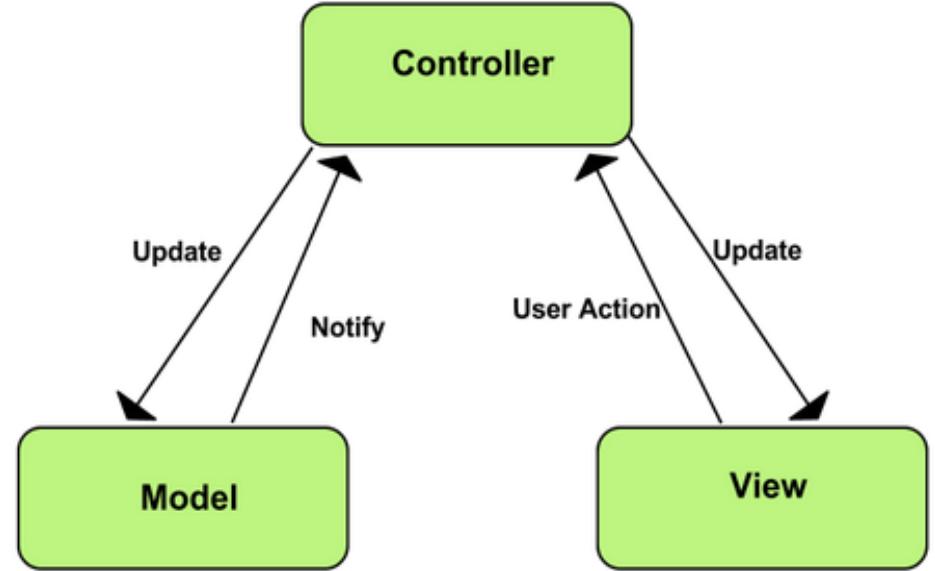
**MVC** = Model View Controller.

# MVC

## M: Model

The model defines how the application's data is stored. It is responsible for managing the data.

- When a model changes, it will **notify** the controller that a change has occurred.
- It responds to the instructions from controller to **update** itself.



## V: View

View is a particular presentation of the data to the users and how users interact with the app. The view is made with HTML, CSS, JavaScript and often templates (e.g. PHP templates).

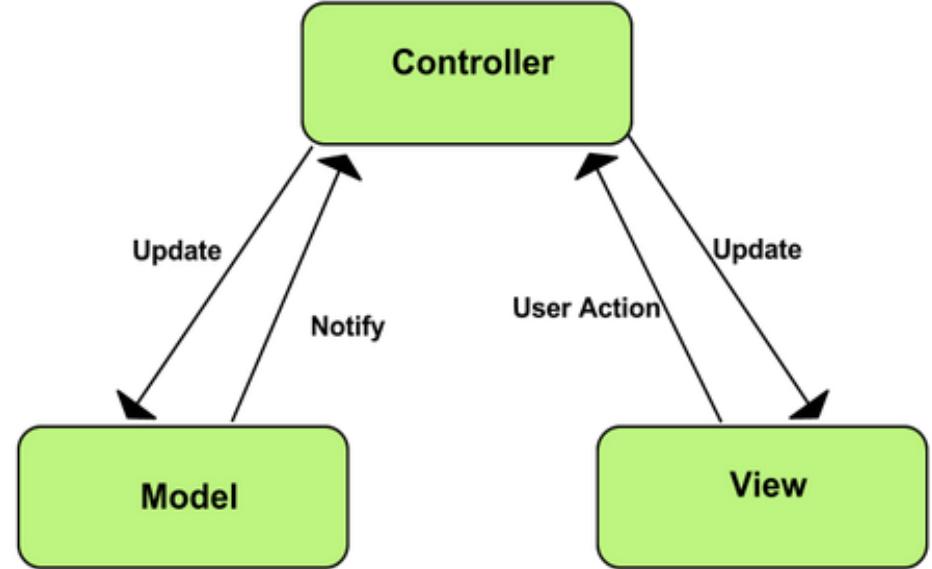
- When a user takes an action, it will **notify** the controller that a change has occurred.
- It responds to the instructions from controller to **update** itself.

## C: Controller

The controller responds to user actions and performs interactions on the data model objects. The controller receives input, validates it, and then performs operations that modify the state of the data model.

# MVC

MVC is popular because it isolates the application logic from the user interface layer and supports separation of concerns. The controller receives all requests for the application and then works with the model to prepare any data needed by the view. The view then uses the data prepared by the controller to generate a final presentable response.



In the early days of the Web, the MVC architecture was mostly implemented on the server-side, with the client requesting updates via forms or links, and receiving updated views back to display in the browser.

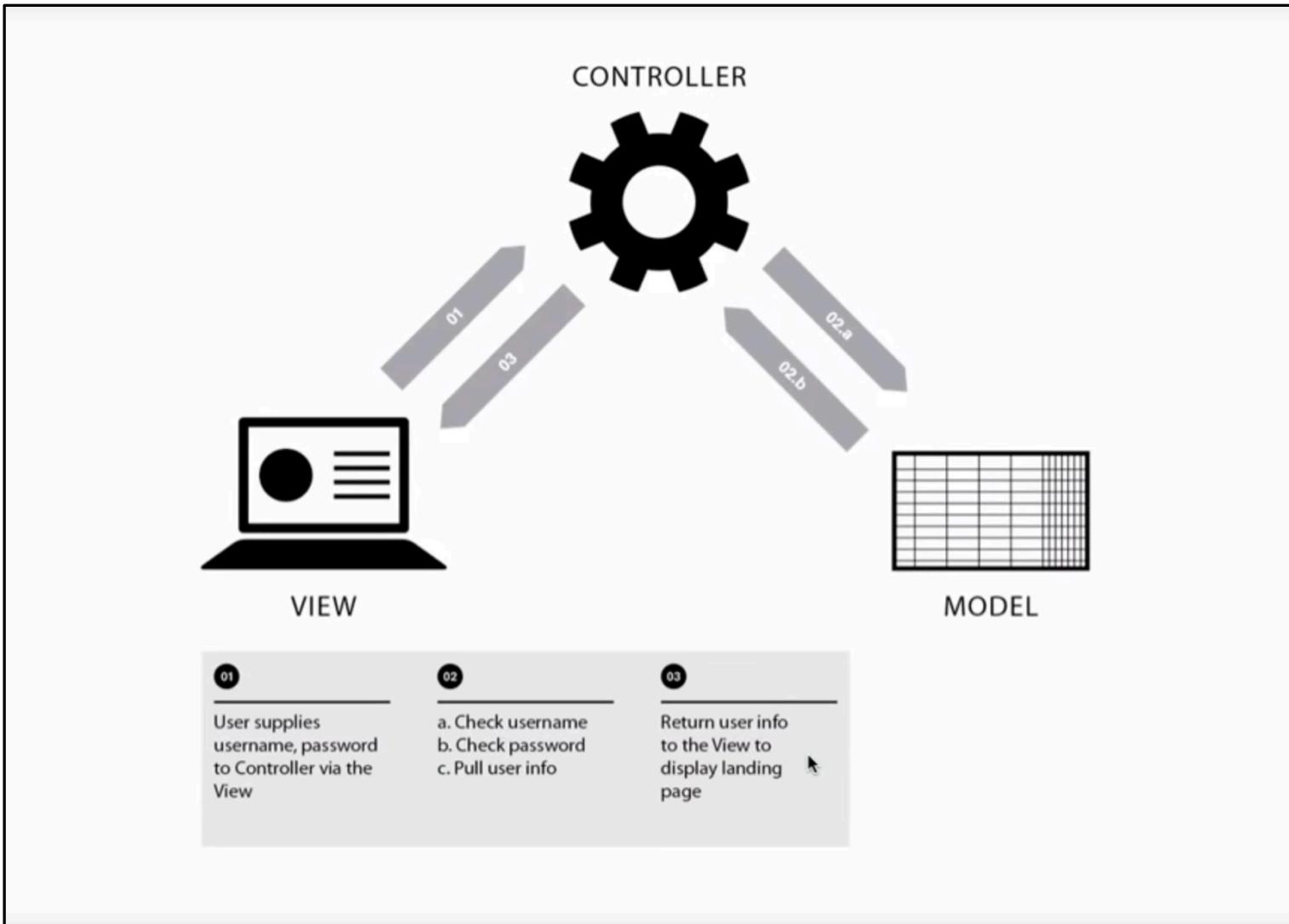
- Popular web frameworks such as Django, Flask, Ruby on Rails all implement the MVC architecture, albeit in slightly different ways.

These days more of the logic is pushed to the client with the advent of client-side data stores, and XMLHttpRequest allowing partial page updates as required.

- Client-side frameworks such as AngularJS, Ember.js and Backbone implement an MVC architecture, but again in slightly different ways.

# MVC

A nice video about MVC with examples: [https://www.youtube.com/watch?v=LiBdzE\\_DJn4](https://www.youtube.com/watch?v=LiBdzE_DJn4)



(ignore the add for the book at the end)

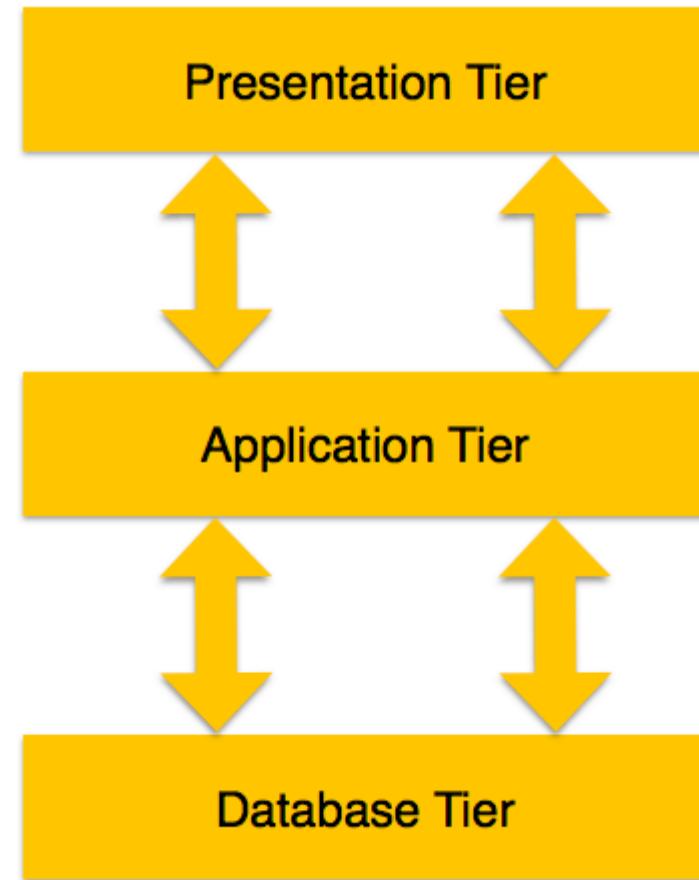
# Three tier architecture

In software engineering, **multitier architecture** (often referred to as **n-tier architecture**) is a client–server architecture in which presentation, application processing, and data management functions are physically separated. The most widespread use of multitier architecture is the **three-tier architecture**.

Multiple-tier architecture is highly modifiable, as almost all its components are independent and can be changed independently.

The added modularity makes it easier to modify or replace one tier without affecting the other tiers.

Separating the application functions from the database functions makes it easier to implement load balancing.

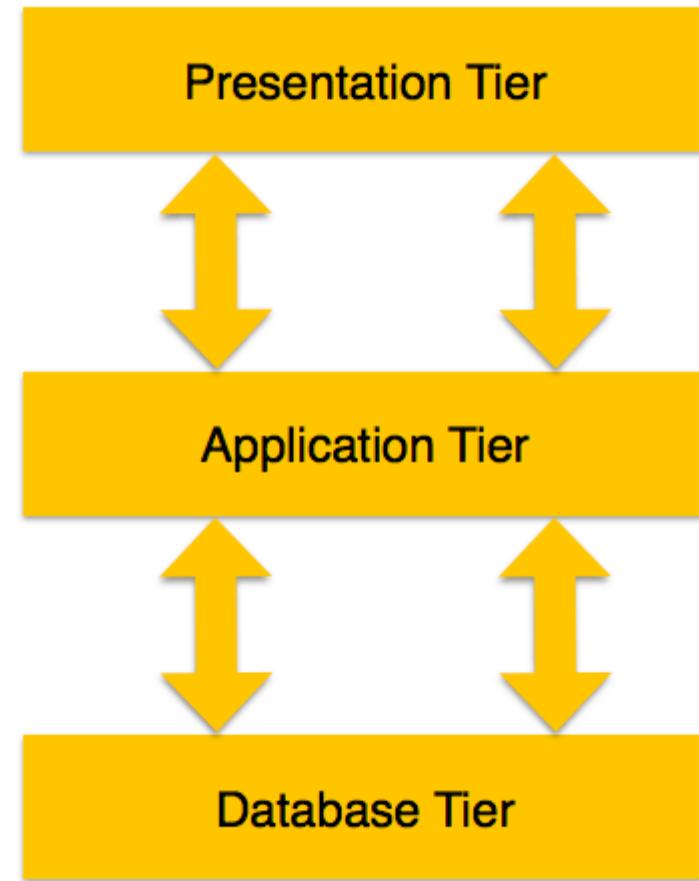


# Three tier architecture

**User (Presentation) Tier** – End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database can be provided by the application. All views are generated by applications that reside in the application tier.

**Application (Middle) Tier** – At this tier reside the application server and the programs that access the database. For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.

**Database (Data) Tier** – At this tier, the database resides along with its query processing languages. We also have the relations that define the data and their constraints at this level.



# JavaScript Frameworks

Project	Ajax	MVC framework	MVC push-pull	I18n & L10n?	ORM	Testing framework(s)	DB migration framework(s)	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)
AngularJS	XHR, JSONP ↗	Yes		I18n and L10n ↗		Karma (unit testing) ↗, Protractor (end-to-end testing) ↗		Content Security Policy (CSP) ↗, XSRF ↗	Templates ↗	Caching ↗	Form validation (front-end) ↗
EmberJS	Yes	Yes		Yes	Ember Data ↗	QUnit ↗			Handlebars ↗		
qooxdoo	Yes	Data binding ↗		I18n ↗		Testrunner ↗					Form Validation ↗
SproutCore	Yes	Yes									
Wakanda	Yes	Yes	Push & Pull		Native Object NoSQL DB ↗	CommonJS Unit Testing ↗ YUI Test Service ↗		Data Security and Access Control ↗		Storage (application.storage, user.storage, SessionStorage) ↗	

Localization and Internationalization

# JavaScript Frameworks

Project	Ajax	MVC framework	MVC push-pull	I18n & L10n?	ORM	Testing framework(s)	DB migration framework(s)	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)
AngularJS	XHR, JSONP ↗	Yes		I18n and L10n ↗		Karma (unit testing) ↗, Protractor (end-to-end testing) ↗		Content Security Policy (CSP) ↗, XSRF ↗	Templates ↗	Caching ↗	Form validation (front-end) ↗
EmberJS	Yes	Yes		Yes	Ember Data ↗	QUnit ↗			Handlebars ↗		
qooxdoo	Yes	Data binding ↗		I18n ↗		Testrunner ↗					Form Validation ↗
SproutCore	Yes	Yes									
Wakanda	Yes	Yes	Push & Pull		Native Object NoSQL DB ↗	CommonJS Unit Testing ↗ YUI Test Service ↗		Data Security and Access Control ↗		Storage (application.storage, user.storage, SessionStorage) ↗	

Object-relational mapping (**ORM**)

Object-relational mapping (**ORM**, O/RM, and O/R mapping tool) in computer science is a programming technique for converting data between incompatible type systems in object-oriented programming languages.

# JavaScript Frameworks

Project	Ajax	MVC framework	MVC push-pull	I18n & L10n?	ORM	Testing framework(s)	DB migration framework(s)	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)
AngularJS	XHR, JSONP ↗	Yes		I18n and L10n ↗		Karma (unit testing) ↗, Protractor (end-to-end testing) ↗		Content Security Policy (CSP) ↗, XSRF ↗	Templates ↗	Caching ↗	Form validation (front-end) ↗
EmberJS	Yes	Yes		Yes	Ember Data ↗	QUnit ↗			Handlebars ↗		
qooxdoo	Yes	Data binding ↗		I18n ↗		Testrunner ↗					Form Validation ↗
SproutCore	Yes	Yes									
Wakanda	Yes	Yes	Push & Pull		Native Object NoSQL DB ↗	CommonJS Unit Testing ↗ YUI Test Service ↗		Data Security and Access Control ↗		Storage (application.storage, user.storage, SessionStorage) ↗	



<https://angularjs.org/>

Miško Hevery, a Google employee, started to work with AngularJS in 2009.

AngularJS version 1.0 was released in 2012.

The idea turned out very well, and the project is now officially supported by Google.

## Why AngularJS?

HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.

## Alternatives

Other frameworks deal with HTML's shortcomings by either abstracting away HTML, CSS, and/or JavaScript or by providing an imperative way for manipulating the DOM. Neither of these address the root problem that HTML was not designed for dynamic views.

## Extensibility

AngularJS is a toolset for building the framework most suited to your application development. It is fully extensible and works well with other libraries. Every feature can be modified or replaced to suit your unique development workflow and feature needs. Read on to find out how.

w3schools: <http://www.w3schools.com/angular/default.asp>

Free course: <http://campus.codeschool.com/courses/shaping-up-with-angular-js/intro>



AngularJS is a library written in JavaScript.

AngularJS is distributed as a JavaScript file, and can be added to a web page with a script tag:

```
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
```

AngularJS extends HTML with **ng-directives**.

- The **ng-app** directive defines an AngularJS application.
- The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data.
- The **ng-bind** directive binds application data to the HTML view.
- The **ng-init** directive initializes AngularJS application variables.
- The **ng-controller** directive defines the controller.

# PHP frameworks – server side

Project	Language	Ajax	MVC framework	MVC push-pull	i18n & L10n?	ORM	Testing framework(s)	DB migration framework(s)	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)	Scaffolding	RAD	Mobility
Banshee	PHP 5	Yes, native and JQuery	Yes	Push	Yes	Yes	No	No	Yes	Yes, XSLT	Yes	Yes	Yes	Yes	Yes, Bootstrap
CakePHP 1 & 2	PHP >= 5.2	Prototype, script.aculo.us, jQuery, jQuery UI, MooTools, MooTools More	Yes	Push	Yes	Active record pattern (CakePHP 1.x), data mapper pattern (CakePHP 2.x)	Unit tests, object mocking, fixtures, code coverage, memory analysis with SimpleTest and Xdebug PHPUnit (cakephp 2.0)	Yes	ACL-based	Themes, layouts, views, elements	Memcache, Redis, XCache, APC, File	Validation, security	Yes	No	?
CakePHP 3	PHP >= 5.4	Any	Yes	Yes, Push & Cells	Yes	ORM, Data Mapper Pattern, SQL Relational Algebra Abstraction Layer	Unit tests, object mocking, fixtures, code coverage, memory analysis with PHPUnit and Xdebug and Continuous Integration via Travis	Yes	CRUD based, ACL-based, Multiple Plugins	Themes, Layouts, Cells, Views, Elements, Plugins for Twig, Bootstrap, etc.	Memcache, Redis, XCache, APC, File	Validation via Contexts (Table (DAO), Entity (VO) & Controller), CSRF Protection	Plugin CRUD ↗	Cake Bake	Mobile Agent Detection, Layouts
Codeigniter	PHP >= 5.2.4	Any	Yes	Push	Mostly [56]	Third party only	Ready for next release	Yes	Yes	Yes	Yes	Yes	No [57]	Yes	Templates
Drupal	PHP	jQuery, jQuery UI, more	PAC	N/A	Yes	Optional module	SimpleTest	Yes	Yes	Yes	Memcache, APC, Varnish, more	Yes	No	No	Yes
Fat-Free Framework	PHP	Any	MVC, RMR	Push-pull	Yes	Data mappers for SQL, MongoDB, Flat-File	Built-in	Yes	Yes	Yes	APC, Memcache, XCache, WinCache, and Filesystem	Yes	No	?	?
FuelPHP	PHP >= 5.3.x	Yes	MVC, HMVC	Push	Yes	Yes	PHPUnit	Yes	Yes, Plugins available	Yes, Plugins available	File, Redis, Memcache, more	Yes	Yes	?	?
Fusebox	PHP	Yes	Not mandatory	Push	No, custom	?	?	?	Multiple plugins available	?	?	via qforms or built in PHP validation	Yes	?	?
Joomla	?	Yes	Plugin	?	?	?	?	?	?	?	?	?	?	?	?
Laravel	PHP ~= 5.5.0	Any	Yes	Push	Yes	Yes	PHP Init	Yes	Yes	Yes	APC, Database, File,	Yes	Yes	Yes	No

# Python Frameworks – server side

Project	Language	Ajax	MVC framework	MVC push-pull	i18n & L10n?	ORM	Testing framework(s)	DB migration framework(s)	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)	Python 3.*
Bottle	Python	-	-	-	-	SQLAlchemy (via plugins)	-	-	-	built-in, Mako, Jinja2, Cheetah	-	-	Yes
CherryPy	Python	-	-	-	-	pluggable	-	-	-	pluggable	-	-	Yes
Django	Python	Yes	Yes	Push	Yes	Yes	Yes	Yes	Yes	built-in, Jinja2, Mako, Cheetah	Yes	Yes	Yes
Flask	Python	Yes	-	-	Yes	SQLAlchemy (via plugin)	via unittest	-	Yes	Jinja2	Yes	Yes	Yes
Pyjs	Python, JavaScript	Yes	Use PureMVC Python version (compiled to JavaScript)	-	Yes	??, no direct data access	-	No	-	-	-	-	No
Pylons	Python	helpers for Prototype and script.aculo.us	controller	Push	Yes	ORM-agnostic	via nose	depends on ORM	-	pluggable: Mako, Genshi, Myghty, Kid, more	Beaker cache (memory, memcached, file, databases)	preferred formencode	No
Pyramid	Python	Yes	Yes	Push	Yes	ORM-agnostic	Yes	depends on ORM	Yes	pluggable: Chameleon, Genshi; Mako, more	Beaker cache (memory, memcached, file, databases)	preferred formencode	Yes
TACTIC	Python	Yes	Yes	Pull	No	Yes	Yes	Yes	Yes	Mako	No	Yes	No
Tornado	Python	See Advanced Async Example implements AJAX ↗	-	-	-	-	-	-	-	-	-	-	Yes ↗
TurboGears	Python	Toolkit-independent, provides support via JSON	Full stack, best-of-breed based	Push	Yes	SQLAlchemy	nose	SQLAlchemy-Migrate ↗	Repoze.what & Repoze.who	pluggable: Genshi, more	Support for memcached, and any WSGI compliant system	ToscaWidgets, utilizing FormEncode ↗	Yes
web2py	Python	Yes	Yes	Push	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Webware	Python	No	Optional	Pull	No	Yes	Yes	No	Yes	Yes	No	No	No



# Flask

web development,  
one drop at a time

[overview](#) // [docs](#) // [community](#) // [snippets](#) // [extensions](#) // [search](#)

Flask is a microframework for Python based on Werkzeug, Jinja 2 and good intentions. And before you ask: It's [BSD licensed!](#)

## Flask is Fun

Latest Version: [0.10.1](#)

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

## And Easy to Setup

```
$ pip install Flask
$ python hello.py
* Running on http://localhost:5000/
```

<http://flask.pocoo.org/>

Django makes it easier to build better Web apps more quickly and with less code.

[Get started with Django](#)

## Meet Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.



Ridiculously fast.

Django was designed to help developers take applications from concept to completion as quickly as possible.

Download latest release:  
[1.9.5](#)

[DJANGO DOCUMENTATION >](#)

## Support Django!



Jannis Leidel donated to the Django Software Foundation to support Django development. [Donate today!](#)

## Latest news

<https://www.djangoproject.com/>

# django

<https://www.djangoproject.com/>

**Django** is a free and open-source web framework, written in Python, which follows the model–view–controller (MVC) architectural pattern.

It is maintained by the [Django Software Foundation](#) (DSF), an independent organization established as a 501(c)(3) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself.

Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include Pinterest, Instagram, Mozilla, The Washington Times, Disqus, and Bitbucket.

The screenshot shows the Django Admin interface for managing users. The top navigation bar includes links for Home, Users, Wikipedian, History, View on site, and Log out. The main title is 'Change user' for a user named 'Wikipedian'. The 'Username' field contains 'Wikipedian' with a note that it must be 30 characters or fewer and contain alphanumeric characters only. The 'Password' field contains a hashed value 'sha1\$040d5\$5e8110f429f463874c2c18' with a note about using the change password form. The 'Personal info' section has empty fields for First name, Last name, and E-mail address. The 'Permissions' section includes checkboxes for Staff status (unchecked), Active (checked), and Superuser status (unchecked). The 'Available user permissions' list includes numerous Django-specific permissions like 'admin | log entry | Can add log entry' and 'auth | user | Can add user'. The 'Chosen user permissions' list is currently empty. Below the permissions is a section for 'Important dates' showing 'Last login' at 2008-08-22 16:19:01 and 'Date joined' at the same time. The 'Groups' section shows the user is part of the 'admin' group. At the bottom are buttons for Delete, Save and add another, Save and continue editing, and Save.

# Questions, Suggestions?

**Please complete the module feedback form.**

Next:

I don't teach on the MSc again this year, but I will be around and am happy to give pointers on issues Web Dev related.