CSCI 1320
Creating Modern Web Applications
Content Management Systems
Brown CS Website
Static Brown CS Website

- Up since 1994
- 5.9 M files (inodes)
- 1.6 TB of filesystem space
Static HTML Generators

- Convert plain-text markup to HTML
  - e.g. Markdown
- Advantages
  - Templating
  - Versioning
    - with git
  - Secure
  - Fast
- Disadvantages
  - Not interactive
  - Must use markup
Static HTML Generators

- Website Meta Language
  - debian.org
  - HTML → HTML
  - Nine passes!
    - C preprocessor
    - HTML macro expansion
    - Embedded Perl
- Sane choices available, too
  - Jekyll
    - Markdown → HTML

```wml
#include <csdept.wml> title="Down For Maintenance"
<h1>Down For Maintenance</h1>
<p><b>We'll be back in just a few minutes...</b></p>
</h1>
</p>
```

An h1 header
============

Paragraphs are separated by a blank line.

2nd paragraph. *Italic*, **bold**, and `monospace`. Itemized lists look like:

* this one
* that one
* the other one
Content Management System

- What does that mean really?
  - Document Management System
  - Enterprise Content Management System
  - Web Content Management System
  - Wiki
  - Blog
  - Digital Asset Management

- CMS
  - Web application for website publishing
One App or Many?

- Monolithic App (+ plugins)
  - WordPress
- Web Framework
  - Multiple apps on one server
- Reverse Web Proxy
  - Multiple apps on multiple servers appear as one site
CMS == Web Application

- Framework
  - Django
- Application
  - WordPress
- Plugin or Extension
  - Slideshow
- Library
  - Jinja2
Reverse Proxy

- Web server directs traffic
- Presents a unified view of separate applications
- Can short-circuit static content
CS Website Applications

- cs.brown.edu
- nginx proxy
- "csdept" django site
- Django-cms
- Zinnia blog
- 49 django apps in all
- Old apache static site
- /web/cs/web
CS Web Server Topology

- How a 404 happens

HTTP → No → HTTP → No → File

No → web proxy → WSGI → No

No → Postgresql

No → apache static server

No → NFS

File? → No → File?
Some Assembly Required

- CMSes based on frameworks can require more setup
  - Separate apps, modules, extensions
  - But they are more flexible
- Monolithic apps are quick to deploy
  - WordPress, of course
Web Content Management System

• Features
  • Templates
  • Roles and permissions
  • In-browser editing
  • Publishing workflow
  • Asset management
  • Versioning
  • Language support

© 2009 Courtoons & David E. Mills
CMS Features – Template Engines

- “V” in MVC
- Separate presentation from application logic
- Render an object or objects
- Template = HTML + some markup
- Break down page design into reusable pieces
- Separate graphic designer from developer chaos
- Lots of choices
  - Jinja2
  - Template Toolkit
  - Django
Django Template Engine

- base.html
  - {% block footer %}{% endblock %}
- page.html
  - {% extends base.html %}
  - {% block content %}stuff{% endblock %}

base.html defines block content
extends
page.html only touches block content
CMS Features – Roles and Permissions

- Authentication
  - Who are you?
  - Email registration, LDAP, Shibboleth, Facebook

- Roles
  - Admin
  - Author
  - Editor
CMS Features – Roles & Permissions

• Authorization
  • What are you allowed to do?
  • Pages have owners
  • Distinct privileges for viewing, creating, editing, publishing, moving, changing ownership, changing permissions

• Hiding Complexity
  • Complex interface for admin
  • Simple interface for authors
    • Don't show me things I can't do
CMS Features – In-browser Editing

- Separate admin or on-page editing?
- Ajax Editors
- Layout and Styling
  - What, not how
- Images and media
  - Upload
  - Asset manager
- Plugins
  - e.g. Google Map
CMS Features – Publishing Workflow

- Content lifecycle
- Draft content
- Notifications
  - Submitted for approval
  - Approved/rejected
- Scheduled publishing
  - Publish Monday @9am
  - Unpublish on Friday @5pm
CMS Features – Asset Management

- Media & documents
  - Images, video
  - Pdf, Word, Spreadsheets
- Avoids duplication
- Metadata
  - Ownership
  - Permissions
  - Copyright
- Search
- Extra step for authors
CMS Features - Versioning

- Oops!
- Page history
  - Shows who did what, when
- Revert to previous page
CMS Features – More Good Things

- Multilingual
  - Multiple versions of site in different languages
- Multi-site
  - Multiple sites running on one server
- Hierarchical pages (tree-like)
  - With hierarchical permissions
- Friendly URLs
  - RESTful
Choices!

- So many to choose from:
  - Java
  - Perl
  - Python
  - Ruby
  - PHP
CMS Market Share

- WordPress 58.8%
  - Joomla! 7.1%
  - Drupal 4.7%
  - Everyone else
QWERTY Keyboard
Ubiquity

- Qwerty designed in 1873
- Optimized for either
  - Avoiding mechanical jams
  - Telegraph operators
Path Dependence

• From economics
  • Making optimal choices
  • History matters because...
  • Switching is hard
• Optimal?
  • English measurement system
  • X86 instruction set
  • PHP
  • Javascript
  • The whole web stack
LAMP

- Linux Apache MySQL Php
- The ubiquitous platform
  - Default on hosting companies everywhere
Why Django-cms?

- Good enough
  - WordPress probably still easiest to use
- Django framework
  - We build our own apps
    - Brown user profiles
    - Courses
    - DIYTube
  - Strong community
  - Secure, ORM, etc...