

Published 2011 by

Acquia, Inc.
25 Corporate Drive, 4th Floor
Burlington, MA 01803

Editor: Heather James

Acknowledgements

Screenshots of the case study websites are used with generous permission of their respective trademark owners. We kindly thank SocialSignal.com, a social media strategy firm, for use of screenshots of their site.

Portions of this manual were originally created by Acquia Training Partners such as Zivtech and Chapter Three.

Feedback

Your feedback is much appreciated. Please don't forget to tell us about your experience.

<http://www.acquia.com/training-feedback>

We're always improving these materials and keeping them up to date. The modules are often updated in between publishing dates. You may find some differences in the user interface, especially discrepancies in screenshots. We appreciate your feedback and will work as quickly as possible to incorporate any corrections.

Appendix

Please download the PDF full appendices available at <http://tinyurl.com/trainingappendix>

In addition to this guide we will also make extensive use of the excellent Drupal.org Theme Handbook as a reference: <http://drupal.org/theme-guide/6-7>

Table of contents

Course overview	4
Schedule	5
Learning environment	6
Modules and files used in this course	12
Using the Demo module	14
Before we start	15
Session 1: Drupal theming fundamentals	17
Unit 1.1 - What is the theme system?	17
Unit 1.2 - What is theming?	19
Unit 1.3 - What are themes?	20
Unit 1.4 - Make a simple theme	27
Unit 1.5 - Add CSS and JavaScript	32
Unit 1.6 - Anatomy of a theme	36
Unit 1.7 - Resetting your site	38
Session review	39
Session 2: Customize with templates	40
Unit 2.1 - Tools for theme developers	40
Unit 2.2 - Introduction to overriding template files	42
Unit 2.3 - PHP basics: Conditional statements	45
Unit 2.4 - Practice extending your theme	46
Unit 2.5 - Theming contributed modules	50
Unit 2.6 - Views template suggestions	53
Session summary	58
Session 3: Theme development process	59
Unit 3.1 - Comparison of processes	59
Unit 3.2 - Identify what we forgot to theme	60
Unit 3.3 - Case study: Social signal	61
Session summary	62
Session 4: Advanced theming introduction	63
Unit 4.1 - Coding standards in Drupal	63
Unit 4.2 - Basic structures in PHP	64

Unit 4.3 - PHP Basics, Functions	65
Unit 4.4 - Using functions in Drupal	66
Unit 4.5 - PHP Basics, Variables	68
Unit 4.6 - Variables in Drupal	69
Unit 4.7 - PHP Basics, Arrays	70
Unit 4.8 - Render arrays - How Drupal creates the HTML string	71
Session Summary	73
 Session 5: Controlling layout through configuration	 74
Unit 5.1 - Site building tips	74
Unit 5.2 - Contributed modules for theming	75
Unit 5.3 - Display suite for custom teaser	79
Unit 5.5 - Panels for custom layouts	84
Unit 5.4 - Context module for site sections	85
Session summary	88
 Session 6: Responsive theme systems	 89
Unit 6.1 - What are sub-themes?	89
Unit 6.2 - What are base themes?	90
Unit 6.3 - A quick demo of the Adaptive system	91
Unit 6.4 - Omega: A responsive theme system	93
Session summary	100
 Session 7: Front-End Security and Performance	 101
Unit 7.1 - Security vulnerabilities	101
Unit 7.2 - Tools for checking front end performance	102
Unit 7.3 - Drupal configuration check for front end performance	103
Unit 7.4 - Optional - Selectively add CSS or JS	104
Session summary	106
 Session 8: Review: How would you do that in Drupal?	 107
 Appendix: Reference materials	 108

Course overview

This course assumes you're an experienced site builder with Drupal. You know how to configure a Drupal site, but you've found limitations with what you can do to customize the look and feel. You do have experience with HTML and CSS. While you don't need to be an expert in PHP, you can copy and paste your way around some customizations.

Critics of Drupal's theme selection may not realize how Drupal is quite unique. It's more difficult to make a "generic" Drupal theme for contribution on Drupal.org. And what they don't know is that making your own custom theme, especially in Drupal 7 is, as you will see, surprisingly easy.

This two day course is an in-depth introduction to front end development with Drupal. In this course you will find out about best practices with configuring Drupal's output, and what you can do to intercept and override output to get the exact customization you need.

The procedures are introduced step-by-step and then reinforced with practice and challenges. Your instructors can help guide you through the process of establishing this new way of working. You'll look at case studies of websites and reflect on your own experience to take the procedures you learn here back into your own work and any new situations you encounter.

Pre-requisites and prior experience

In order to get the most out of this course you should have prior experience in site building with Drupal. A combination of Acquia's Drupal in a Day, and Site Builder two day course would be ideal preparation.

- Drupal related site configuration.
 - Content creation and maintenance.
 - Site building.
- HTML and CSS skills are essential.
- Familiarity with both Javascript and PHP would be helpful but not necessary.

What you will be able to...

After you complete this course, you will be able to...

- Articulate theming concepts and terms.
- Make a basic custom theme from scratch.
- Make sub-themes of a popular base theme system.
- Understand inheritance and overriding templates.
- Manipulate mark-up at the theme layer.
- Employ best practices for creating sustainable and secure themes.
- Tune your site for the best front end performance.

Schedule

These start and end times are a guideline based on a 9-4:30pm schedule. Depending on your start and finish time they may be different.

Day 1

You should have a working installation of Drupal with a fresh site installed.

Start	Duration	Session	Description
On arrival		Before we start	Look for inspiration and form questions
9:00	0:15	Welcome	What we'll do today
9:15	1:00	Session 1	Drupal theming fundamentals
10:15	0:15	Break	Time for a break
10:30	1:30	Session 2	Customize with templates
12:00	1:00	Lunch	If it's nice outside, go for a walk.
1:00	1:30	Session 3	Theme development process
2:30	0:15	Break	Afternoon break
2:45	1:45	Session 4	Advanced Theming

Day 2

You will need a fresh Drupal site to work with for Day 2.

Start	Duration	Session	Description
On arrival		Before we start	Look for inspiration and form questions
9:00	1:15	Session 5	Controlling layout through configuration
10:15	0:15	Break	Time for a break
10:30	1:30	Session 5	Controlling layout (continued)
12:00	1:00	Lunch	If it's nice outside, go for a walk.
1:00	1:30	Session 6	Responsive theme systems
2:30	0:15	Break	Afternoon break
2:45	0:30	Session 7	Front end Security and Performance
3:15	1:45	Session 8	Review and Discussion

Learning environment

Exercise: Get your environment set up

A. Set up Dev Desktop

1. You need a local installation of the Acquia Dev Desktop. This includes Drupal, Apache, MySQL and PHP.
2. Go to <http://www.acquia.com/downloads> to download the Drupal 7 version of Dev Desktop for your operating system.

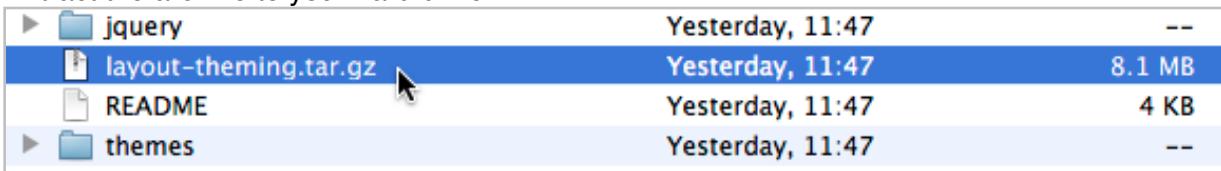
B. Download the files

1. Go to <https://github.com/acquia/training/downloads>
2. Click "Downloads" to download the archive available. Follow the links to "Download as tar.gz".

The screenshot shows a GitHub repository page for 'training /'. The 'Downloads' tab is selected. The page displays a list of files with their names, ages, messages, and history links. The files listed are:

name	age	message	history
jquery	4 months ago	Added jessebeach's jquery plugin template to the repo as a submodule. [jessebeach]	
themes	a day ago	Renaming session 1 themes. [nearlythere]	
.DS_Store	a day ago	Adding in the session themes [nearlythere]	
.gitmodules	4 months ago	Added jessebeach's jquery plugin template to the repo as a submodule. [jessebeach]	
README	4 months ago	edited the readme [jessebeach]	
layout-theming.tar.gz	a day ago	Adding in the session themes [nearlythere]	

3. Extract the archive to your hard drive.

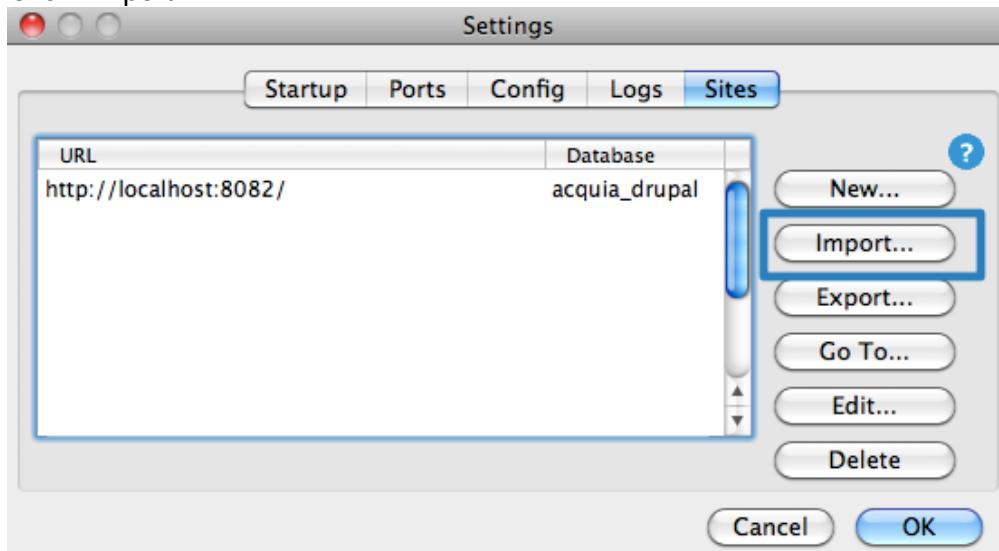


4. Extract the layout-theming.tar.gz. This makes a 'www' folder. This can sit anywhere on your hard drive. **It should not be placed inside your "acquia-drupal" folder.**

C. Import into Dev Desktop

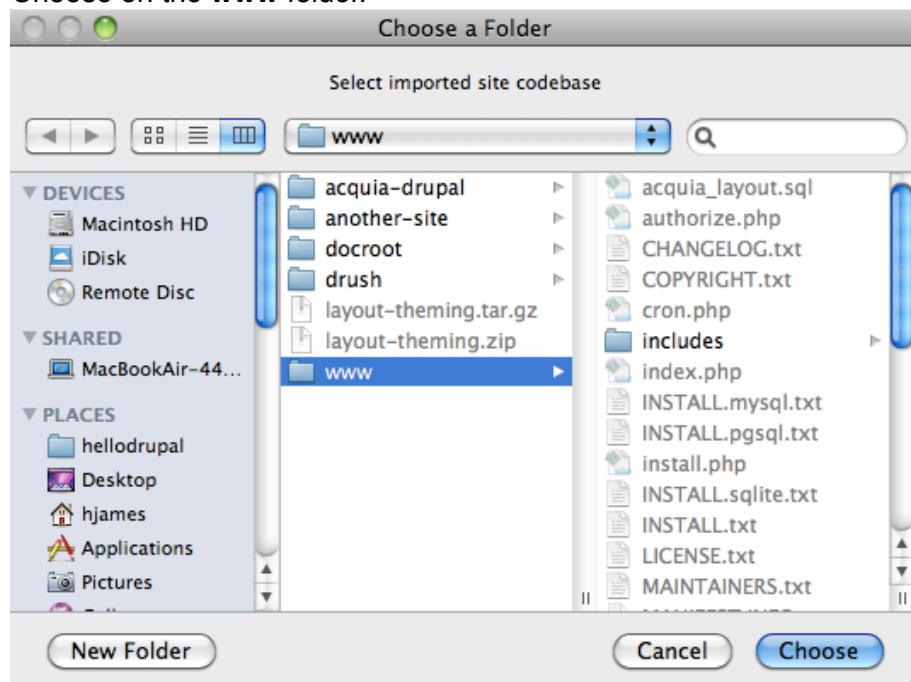
1. Open Acquia Dev Desktop Control Panel.
2. Click the "Settings..." button then "Sites" tab in the Settings dialog.

3. Click "Import...".

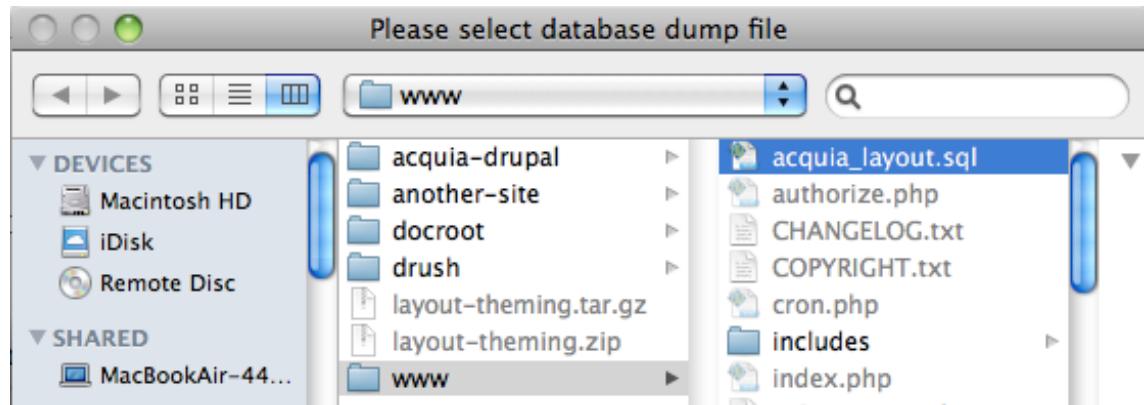


4. Fill in the form with the following settings:

- **Site path:** Browse to the folder you just created when you extracted the archive. Choose on the **www** folder.



- **Database > Dump file:** Select the SQL dump "acquia_layout.sql" located in the extracted folder.



- **New DB name:** Choose whatever name you like. **it is recommended that you choose a name that relates to the course.
- **Subdomain:** Choose whatever name you like. **it is recommended that you choose a name that relates to the course.

Import site

Codebase:

Site path: /Users/hjames/Sites/www

Subsite: default

Database:

I have a MySQL database dump file for my site
 Create new database
 Import directly from MySQL server

Dump file: /Users/hjames/Sites/www/acquia_layout.sql

New DB name: themedb

Domain:

Server: localhost

Subdomain: learntheming

URL path:

Site URL: http://learntheming.localhost:8082/

5. Click **Import**.

D. Launch and log in to the site

1. Go back to your site listing and click "Go to..." to launch your site. Or go to your browser and type the Site URL as it appears in settings. For example <http://learntheming.localhost:8082/>
2. The admin credentials are username: admin / password: admin

3. You will now see your working sandbox:

The screenshot shows the Acquia Drupal dashboard. At the top, there's a navigation bar with links like 'Dashboard', 'Content', 'Structure', 'Appearance', 'People', 'Modules', 'Configuration', 'Reports', 'Help', 'Subscription not active' (with a red exclamation mark icon), 'Hello admin', and 'Log out'. Below the navigation is a search bar with 'Add content' and 'Find content' buttons. The main header features the 'acquia drupal' logo and the text 'My Acquia Drupal Site'. A 'Home' button is visible. A yellow callout box in the center says: 'Get a free 30 day trial of Drupal support, enhanced content search, comment spam blocking and more. If you have an Acquia Network subscription, connect now. You can turn this message off by disabling the Acquia Network modules too.' To the right, there's a 'My account' link and a 'Log out' button. On the left, there's a search bar and a search icon. On the right, there's a 'Reset demo' button and a note about an 'Active snapshot: training-default' with a 'Reset now' button.

Exercise: Set up your Acquia account

When your first login you will see your Acquia account prompt.

! Get a free 30 day trial of Drupal support, enhanced content search, comment spam blocking and more. If you have an Acquia Network subscription, [connect now](#). You can turn this message off by disabling the Acquia Network [modules](#) too.

If you have an account, you can connect it. Alternatively, you can disable the modules and the prompt will disappear.

What is the Acquia Network?

In addition to the use of support services; you get free access to services such as Drupalyze.me, Mollom, Visual Website Optimizer and more. You can also get access to site analysis tools such as Acquia Insight and the SEO grader. This is a great way to find "gotchas" and identify direct solutions to improve your site.

Find out more <http://www.acquia.com/products-services/acquia-network/>



Option 1 - Connect your site to the network

1. If you don't have a network subscription, sign up at www.acquia.com/30-free-trial
2. Login to Acquia Network, and visit the Subscription page at: <https://network.acquia.com/network/dashboard/subscription>

- Locate the Identifier and Key details for the selected subscription.

The screenshot shows a user interface for managing Acquia Network subscriptions. At the top left, there is a dropdown menu labeled "Site: Select Site". Below it, the heading "Details for all sites" is displayed in blue. A table lists one subscription entry:

Subscription	Identifier	Key
SubscriptionName (Expires: 12/31/2012)	7ba417e4e692c56806fbfa51738658	690441d0de31d996cea4e6838e4bf345 Add/Edit ▾ Upgrade ▾

At the bottom right of the table area is a button labeled "Add a new subscription".

- Copy these details so you can place them into the Acquia Network settings for your new site.
- Go to Configuration > System > Acquia Network settings or click "Connect now"
- Paste in Identifier and Key.
- Click **Save keys**.

The screenshot shows the "ACQUIA NETWORK KEYS" configuration page. It contains two input fields: "Identifier *" with the value "7ba417e4e692c56806fbfa517386586" and "Key *" with the value "690441d0de31d996cea4e6838e4bf34". Below each field is a link to "Copy from your subscriptions overview on the Acquia Network". At the bottom are two buttons: "Save keys" and "Clear keys".

If you get lost along the way, visit the documentation pages here: <http://network.acquia.com/documentation/getting-started/introduction>

You'll need to run cron to ensure Acquia is connected. The network service begins to provision a search slice for you soon as you request it, though it may take a few moments. You may get an error such as "Your site was unable to contact the Apache Solr server." This error will go away after a few minutes.

Option 2 - Disable Acquia connector

- Go to Modules listing and locate the Acquia Network Connector group.
- Disable the Acquia Site information module.

3. Then disable the Acquia agent.

▼ ACQUIA NETWORK CONNECTOR

ENABLED	NAME	VERSION	DESCRIPTION	OPERATIONS
<input checked="" type="checkbox"/>	Acquia agent	7.x-1.3	Allows Drupal to securely communicate with the Acquia Network, and checks for updates to Acquia Drupal. Required by: Acquia site information (enabled), Acquia search (disabled)	
<input type="checkbox"/>	Acquia search	7.x-1.3	Provides integration between your Drupal site and Acquia's hosted search service. Requires: Acquia agent (enabled), Apache Solr framework (disabled), Apache Solr search (disabled), Search (enabled)	
<input checked="" type="checkbox"/>	Acquia site information	7.x-1.3	Sends a site profile including Acquia Drupal version, platform and module information; number of posts, comments and users; as well as file hashes and file metainformation on files in the system to the Acquia Network to facilitate support and update notifications. Requires: Acquia agent (enabled)	? Help

Modules and files used in this course

In addition to the modules included with Acquia Drupal, these are modules used in this course.

- Context <http://drupal.org/project/context>
- Demo <http://drupal.org/project/demo>
- Developer (Devel) <http://drupal.org/project/devel>
- Display Suite <http://drupal.org/project/ds>
- Style Guide module <http://drupal.org/project/styleguide>
- Theme Developer http://drupal.org/project/devel_themer
- Views <http://drupal.org/project/views>

Exercise: Code samples and files

At this stage, you don't have to download any files, simply familiarize yourself which what is available and where it is. **You can refer to these instructions later. They should not be completed before the class starts.**

A. Locate the code samples

1. Go to: <https://github.com/acquia/training/tree/master/themes>
 - You do not need a GitHub account to download the files, or view them within the browser.
2. You will see a number of themes. They will be referred to throughout the course.

The image shows a GitHub repository interface and a corresponding file tree diagram. On the left, a table lists theme files with columns for name and age. A row for 'mytheme' is highlighted with a blue border. On the right, a hierarchical file tree shows the directory structure of the 'mytheme' folder, which contains subfolders for 'Final', 'Session 1', 'Session 2', and 'Session 4', each containing multiple theme files.

name	age
..	
basic	November 16, 2011
bullfrog	October 31, 2011
cornflower	January 16, 2012
jasmin	November 16, 2011
minimum	November 16, 2011
myomega	December 08, 2011
mytheme	2 minutes ago
responsive	September 08, 2011
starter	December 02, 2011
.DS_Store	2 minutes ago

File tree diagram:

- mytheme
 - Final
 - Session 1
 - mytheme_1_3
 - mytheme_1_4
 - Session 2
 - mytheme_2_0
 - mytheme_2_2
 - mytheme_2_4
 - mytheme_2_5
 - mytheme_2_6
 - Session 4
 - mytheme_4_4
 - mytheme_4_6
 - mytheme_4_8

B. Restarting "mytheme" to catch up

Throughout the first activities you'll be building a theme called "mytheme". Snapshots of the stages of this theme are available for reference, the numbers of the folders refer to the unit session and unit numbers.

For example, imagine you're just about to start **Session 1, Unit 1.5** and you've made a number of experimental changes. If you want yours to match the example in the manual, follow these steps.

1. Disable your current "mytheme" in your Appearance theme list.
2. Remove that folder from your sites/all/custom folder. Place it somewhere for reference later, or delete it.
3. Copy the entire folder **mytheme_1_4** and paste it into your sites/all/custom/ folder.
4. Return to your Appearance theme list and enable the "My Theme" option. C. Example themes for reference

C. Sample theme files for reference

You will not need to download these before we start. Just familiarize yourself with the options.

In this course you will make a few themes, both from scratch and from base theme systems. The Drupal 7 underlying core theme system is a good starting point. For reference you can use these themes as a starter.

Locate the Minimum, Basic and Starter themes on Github:

<https://github.com/acquia/training/tree/master/themes/>

- **Minimum** has the basic requirements of a theme.
- **Basic** shows a theme with some customizations.
- **Starter** shows a theme with a number of additional files.
- **Cornflower** and **Jasmin** show a base theme and child theme set up.

You can use these as reference or as starting points for experimentation.

Using the Demo module

If you'd like to save the current version of your site before any changes in configuration you can save a snapshot of your site. Later, you can reset your site to a certain stage.

Exercise: Using snapshots

A. Create a snapshot

1. Go to Structure > Snapshots. Click **+ Create snapshot**
2. Give it a descriptive name, in this case `theme_session_start`.

The screenshot shows the 'Create snapshot' form. At the top, there's a 'Name *' field containing 'theming_session_start'. Below it is a note about allowed characters: 'Allowed characters: a-z, 0-9, dashes ("-"), underscores ("_") and dots.' Under 'Description', there's a text area with the placeholder 'This is the fresh site with no configuration changes.' A note below says 'Leave empty to retain the existing description when replacing a snapshot.' There's also a checkbox for 'Use as default snapshot for cron runs'. At the bottom is a 'Create' button.

3. Click **Create**.

B. Reset snapshot

1. Click the "Reset" tab.
2. Select the snapshot you wish to revert to. Be careful. You will lose ALL changes you've made since your last snapshot.
3. Click the **Reset** button. Check your site it's all fixed to the last version.

The screenshot shows a confirmation dialog asking 'Are you sure you want to reset the site?'. Below it, the 'Snapshot *' field is set to 'theming_session_start (Wed, 02/08/2012 - 05:53, 834.31 KB)'. A warning message in a yellow box says 'This action cannot be undone.' Below that, a note says 'Overwrites all changes that made to this site since the chosen snapshot.' At the bottom are 'Reset' and 'Cancel' buttons, with 'Reset' being highlighted.

Before we start

Visit <http://drupalshowcase.com/> to look for inspiration. Find an example site that makes you wonder "HOW did they do that?" During the course, see what you can figure out on your own. You can use these examples in the reviews and discussions.

The screenshot shows the homepage of the Drupal Showcase website. At the top, there's a navigation bar with links for Home, Categories, Industries, and About. Below the navigation, a main headline reads "Drupal powers sites in every industry". To the right of the headline is a cluster of blue circular icons representing various industries like technology, finance, education, and healthcare. Below the headline, a sub-headline says "From manufacturing to retail to non-profits to government, Organizations around the world are creating extraordinary web experiences with Drupal." A horizontal navigation bar below the sub-headline features logos for MTV, ESTEE LAUDER, MIT MEDIA LAB, and PINK. On the left side, there's a section titled "Browse by Industry" with a grid of industry categories and their counts: Architecture (6), Consumer Products (39), Energy (6), Financial Services (145), High Tech (78), Libraries (3), Consulting Services (7), Cultural (31), Entertainment (164), Government (250), Hospitality (11), Life Sciences (2), Consumer Electronics (16), Education (131), Fashion (13), Health Care (238), Legal (6), and Manufacturing (15). On the right side, there's a yellow button labeled "Login or register to submit a Drupal site", a "My Account" link, and a "ADD A SITE" link. Social media icons for YouTube, Facebook, and Twitter are also present.

Example case study: SocialSignal.com



This is a website developed in Drupal. We can use this as an example of a typical project.

Multiple roles- which role do you play?

This site involved the work of multiple people, and Social Signal describe the process on their site. Read more: <http://www.socialsignal.com/dearsosi/whos-behind-social-signals-new-look>

On a small team, you may play one or more of these roles described below. On a large team, you might have multiple people in each role. In any case, it's helpful if each team member understands what each other member needs to accomplish. See the description of their project, and identify if you have similar roles on your team.

- **Site builder:** "Our new site was built in-house by Aaron Pettigrew, with a little help from one of our favourite Drupal programmers, Khalid Baheyeldin..."
- **Content manager:** "The entire Social Signal team worked together to create and organize the new content you'll find on the site..."
- **Visual designer:** "Social Signal's new logo and web design were created by Duoh's Veerle Pieters... That accounts for nifty flourishes on our site like overlapping tabs and swanky breadcrumbs."
- **Drupal themer:** "Veerle's design was converted to a Drupal theme by Courtney Miller of floatleft... Time and again, Courtney helps us prove that a Drupal site doesn't have to look, well, "drupally" -- and we think she's done it again for us!"

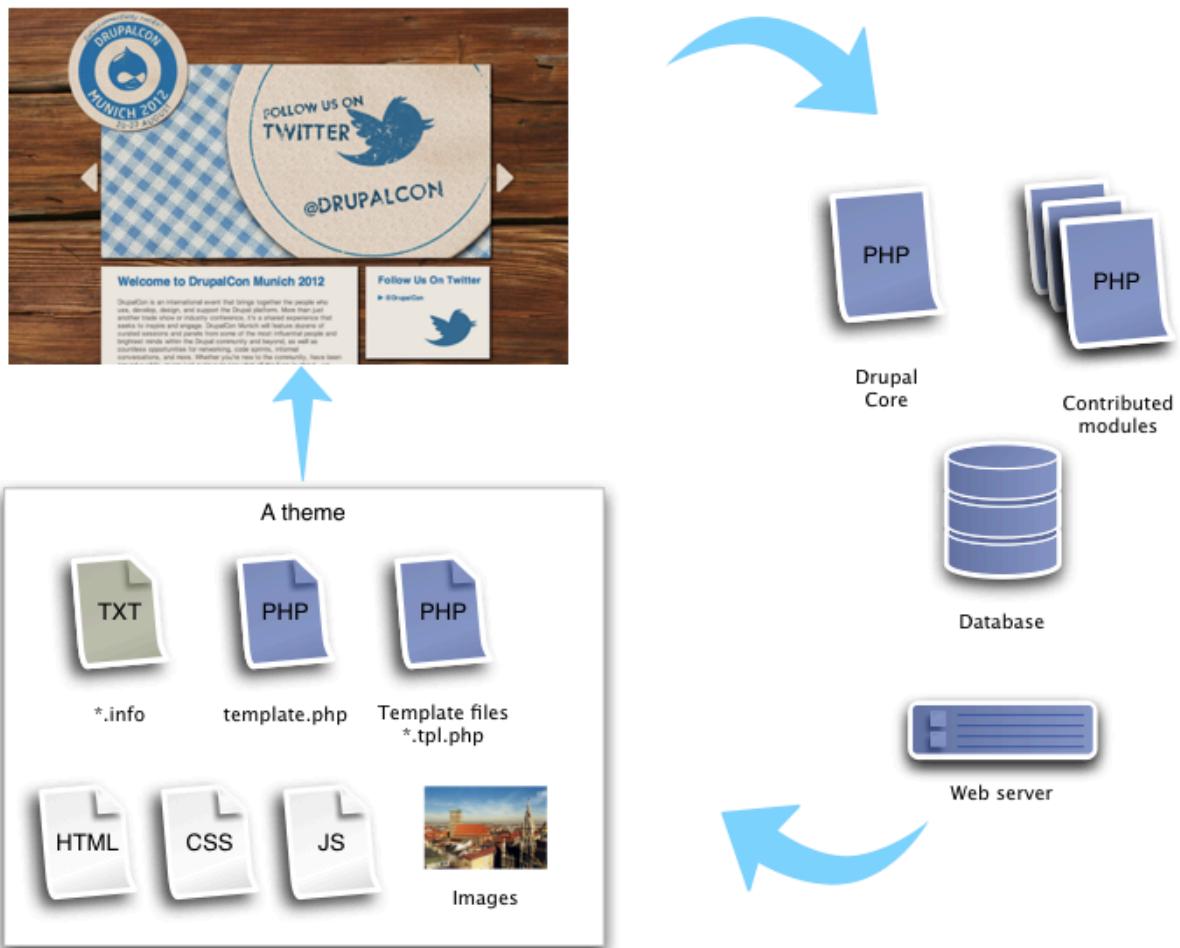
Session 1: Drupal theming fundamentals

In addition to this guide we will also make extensive use of the excellent Drupal.org Theme Handbook as a reference: <http://drupal.org/theme-guide/6-7>

Unit 1.1 - What is the theme system?

The theme system in Drupal allows you to override HTML and CSS created by Drupal core and contributed modules. The theme system controls how a Drupal site looks, and is completely separate from the data, and to a large extent is separate from the management of display of content. For example:

Content type and display	Views	Theme
<ul style="list-style-type: none">TitleBodyImage field as thumbnail	<ul style="list-style-type: none">Display 3 images in a gridOutput as a blockAdd classes	<ul style="list-style-type: none">What color should it be?Border around the image.Padding around the text.



The chain of command

There is a chain of command in the types of languages available used in the theme layer.

PHP > HTML > CSS + Javascript

Each language (or scripting language) has its own purpose.

CSS	The look of the site
HTML (aka markup)	Creation of the desired markup around the content and scaffolding for the CSS and JS
JavaScript	Events and animations that happen within one page
PHP	Logic and database connection

It is important to use each language for its intended purpose, and to follow the chain of command and not attempt to use, for example, JavaScript to accomplish a CSS task, or to use CSS to cover up something that should be handled in markup or PHP. Violating this command chain for momentary convenience tends to cause problems. Another way to think about it is in terms of each level adding enhancement, as the site should still function as the outer-most layers are stripped away.

Unit 1.2 - What is theming?

Themers are developers with super skills to bridge between the visual design goals and the logical layer of programming. Unlike other systems a "blank" theme in Drupal will still render a minimum amount of markup and CSS to give some basic structure to your website. Much of the job of a theme developer and designer is to override these basic settings.

What is theming?

- A plan for a visual design. Perhaps done in an application like Fireworks or directly in static HTML/CSS in the browser.
- Creating CSS to control the layout, positioning and look of your site.
- Alter markup output by Drupal through both configuration of Drupal and editing template files in PHP/HTML.

Skills for themers and front end Developers in Drupal

Depending on the size of your team, you may find yourself either making up the visual design yourself in the browser; or interpreting a design made in Adobe Photoshop by a colleague. To excel at Drupal theming, these related skills are required:

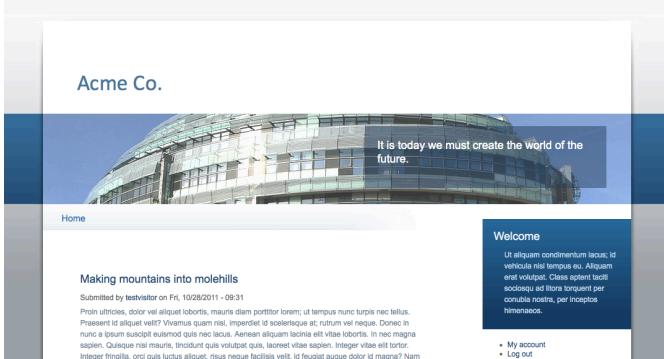
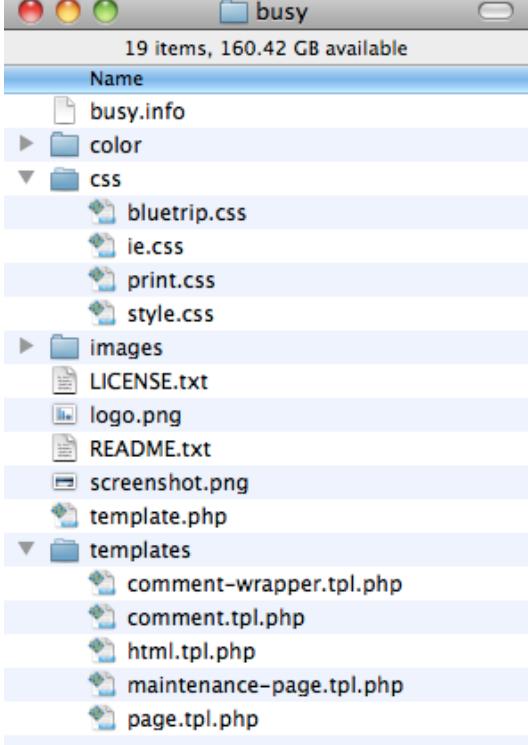
CSS	Advanced
Drupal Site-building	Intermediate
HTML	Advanced
PHP	Beginner

Unit 1.3 - What are themes?

From a technical point of view, a theme is a collection of text files written in HTML, CSS, JavaScript, and PHP code. They often contain images as well. Themes are similar to modules in that they can be downloaded into a themes directory, and enabled via the Appearance admin page. This means, with a few small tweaks you can add a new theme and completely adjust the look of your site while keeping your content intact.

An example theme

Here is the Busy theme <http://drupal.org/project/busy> installed on a demo site. The folder contains CSS files, images and specific templates.

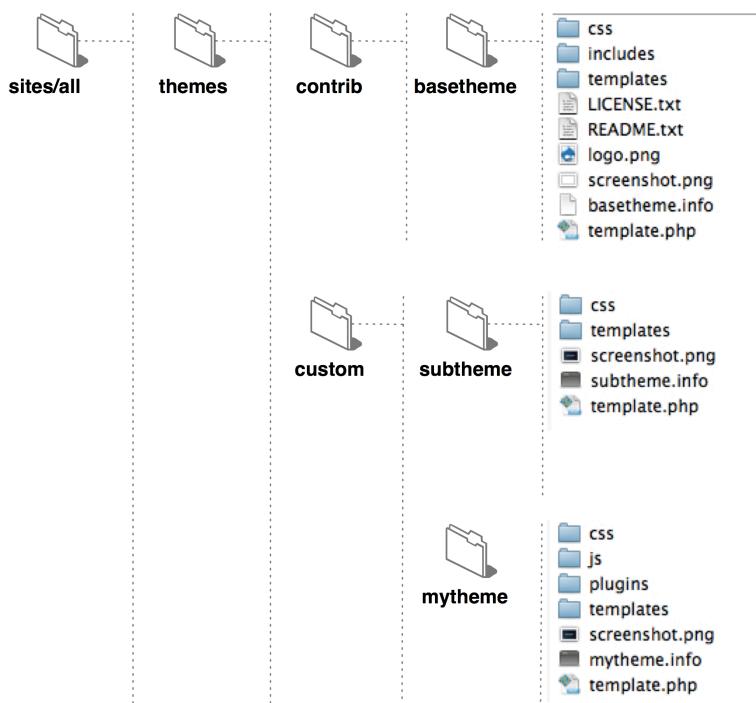
Busy theme screenshot	Files																				
<p>http://drupal.org/project/busy</p> 	<p>sites/all/themes/contrib/busy</p>  <table border="1"><tr><td>Name</td></tr><tr><td>busy.info</td></tr><tr><td>color</td></tr><tr><td>css</td></tr><tr><td> bluetrip.css</td></tr><tr><td> ie.css</td></tr><tr><td> print.css</td></tr><tr><td> style.css</td></tr><tr><td>images</td></tr><tr><td> LICENSE.txt</td></tr><tr><td> logo.png</td></tr><tr><td> README.txt</td></tr><tr><td> screenshot.png</td></tr><tr><td> template.php</td></tr><tr><td>templates</td></tr><tr><td> comment-wrapper.tpl.php</td></tr><tr><td> comment.tpl.php</td></tr><tr><td> html.tpl.php</td></tr><tr><td> maintenance-page.tpl.php</td></tr><tr><td> page.tpl.php</td></tr></table>	Name	busy.info	color	css	bluetrip.css	ie.css	print.css	style.css	images	LICENSE.txt	logo.png	README.txt	screenshot.png	template.php	templates	comment-wrapper.tpl.php	comment.tpl.php	html.tpl.php	maintenance-page.tpl.php	page.tpl.php
Name																					
busy.info																					
color																					
css																					
bluetrip.css																					
ie.css																					
print.css																					
style.css																					
images																					
LICENSE.txt																					
logo.png																					
README.txt																					
screenshot.png																					
template.php																					
templates																					
comment-wrapper.tpl.php																					
comment.tpl.php																					
html.tpl.php																					
maintenance-page.tpl.php																					
page.tpl.php																					

File organization

A theme should be placed in an appropriately named directory within the themes directory, which itself lives within the sites directory of your Drupal site (sites/all/themes or sites/example.com/themes). Many other files can be added if desired.

On the file level, each theme is a directory named uniquely with a *machine name* (underscores and letters only). This name should not match the name of any module used on the site. Custom and contributed themes belong in your site's /sites/all/themes directory (you'll need to make that directory), and core themes, which should not be altered, are in the /themes directory.

A recommended approach to file organization:



Finding themes

You can search for themes at <http://drupal.org/project/themes>.

- Use the filters to narrow down your search.
- Themes are not backward compatible.
- The quality and methods employed in contributed themes varies widely. Always try out on a test site.
- If you require RTL support, you will find less RTL supporting Drupal themes. Though they may not indicate RTL support, be sure to check.

Download & Extend

Download & Extend Home Drupal Core Modules Themes Translat

16 Themes match your search

Filter by compatibility: 7.x
Status: Full projects
Search Themes: RTL
Sort by: Most installed

7.x
8.x
7.x
6.x
5.x
4.7.x

Search

Theme availability

At the time of writing there are 221 compatible Drupal 7 themes. And 2715 compatible Drupal modules. Does this mean theming is harder than module development? In fact, it indicates that extending Drupal is actually quite easy. However, making a "generic theme" for contribution is hard. As you will see later, site building and configuration are a big part of theming.

Terminology: Types of themes

- **Core themes:** Drupal core comes with a few themes. These are suitable for very basic sites. They should not be changed (hacked) nor used as a base for a custom theme.
- **Contributed themes:** <http://drupal.org/project/themes> These are free themes that have been contributed back to the Drupal Community. They are not part of any official release and may not have optimized functionality for your purposes.
- **Custom themes:** Most larger sites require a custom look and feel. These are often created as sub-themes of a *starter or base theme*.
- **Premium themes:** Pre-made themes available for purchase. (e.g., fusiondrupalthemes.com or rockettheme.com) These often have an interface for site-specific customizations and commercial support.
- **Starter themes:** A base theme with a basic layout and little styling designed to be used as a starting point for a custom sub-theme. It is typically much easier to create a custom theme from a starter theme than from scratch.
- **Administration themes:** Themes that are displayed only in administration sections of a site. Some sites use a separate admin theme that is more optimized for administrative pages (e.g. has a wider width for showing big forms).



Make a snapshot using the Demo module!

Before this first exercise, you might like to create a snapshot. Refer to the instructions for "Using the Demo module"

Exercise – Step-by-step install a theme

Let's learn what a theme is by installing and enabling a new theme on our site. Use these instructions for enabling all themes you find.

A. Download a theme

4. Choose a theme from drupal.org/project/themes and download it. Make sure the theme's version compatibility matches your version of Drupal core (7.x). In this example, we're going to use the Busy theme. <http://drupal.org/project/busy>
5. Locate the appropriate download available from the project page. Download.

Busy

[View](#) [Version control](#) [Revisions](#)

Posted by [stBorchert](#) on January 17, 2010 at 2:57pm

Busy is a theme targeted mainly at corporate sites. See some screenshots over here: <http://drupal.org/node/695292#comment-2519468>

To do

- support Color module
- improve Color module support

Busy is sponsored by [undpaul](#)

Project Information

Maintenance status: [Actively maintained](#)

Development status: [Under active development](#)

Reported installs: 1883 sites currently report using this theme. [View usage statistics](#).

Last modified: January 30, 2011

Downloads

Version	Downloads	Date	Links
7.x-1.4	tar.gz (231.27 KB) zip (247.78 KB)	2011-Feb-01	Notes



Maintainers for Busy

[eigentor](#) – 6 commits
last: 10 weeks ago, first: 1 year ago
[aschiwi](#) – 25 commits
last: 18 weeks ago, first: 1 year ago
[stBorchert](#) – 21 commits
last: 38 weeks ago, first: 1 year ago

[View all committers](#)

[View commits](#)

Issues for Busy

To avoid duplicates, please search before submitting a new issue.

[Advanced search](#)

All issues

26 open, 55 total

Bug reports

6 open, 14 total

B. Place in your theme folder

1. Create a contrib theme folder so you have sites/all/themes/**contrib**
2. Unpack the download and move the directory to /sites/all/themes/contrib in your codebase.

C. Enable the theme

1. Go to Appearance and locate the Busy theme under the "Disabled Themes" section.

DISABLED THEMES



Busy 7.x-1.4

The ultimate Drupal business theme.

[Enable](#) | [Enable and set default](#)



Stark 7.4

This theme demonstrates Drupal's default HTML markup and CSS styles. To learn how to build your own theme and override Drupal's

[Enable Busy as default theme](#) [fault code, see the [Theming Guide](#).

[Enable](#) | [Enable and set default](#)

2. Click **Enable and set as default**.

- Keep in mind, you can have multiple themes "enabled" but not active.

3. View your site. You should see some changes.

4. If you have time, try another theme.

Exercise – Explore the theme

A. Configure the theme settings

1. Go to Appearance and click on the tabs per enabled theme. Some themes will come with more settings, such as color options. Compare the themes you have enabled.

The screenshot shows the 'Appearance' configuration page. At the top, there are tabs for 'LIST', 'UPDATE', and 'SETTINGS'. Below the tabs, the 'Global settings' tab is selected. The main content area is titled 'TOGGLE DISPLAY' and contains a list of items to enable or disable. Most items have checkboxes checked, except for 'Logo' and 'User verification status in comments' which are unchecked.

2. In the Global settings, under Toggle display,
 - De-select "Logo",
 - De-select "User verification in comments".
3. Click **Save configuration**.

B. Add a block.

1. Go to Structure > Blocks. Click "+ Add block". Configure with these settings:
 - Block description: Static welcome block
 - Title: Welcome
 - Body: A sample sentence.
2. Scroll down to the **Region settings**. Notice that only the themes you have enabled, whether they are active or not, are available in the block system. For your current theme, select "Sidebar first".

REGION SETTINGS

Specify in which themes and regions this block is displayed.

Busy (default theme)

Sidebar first

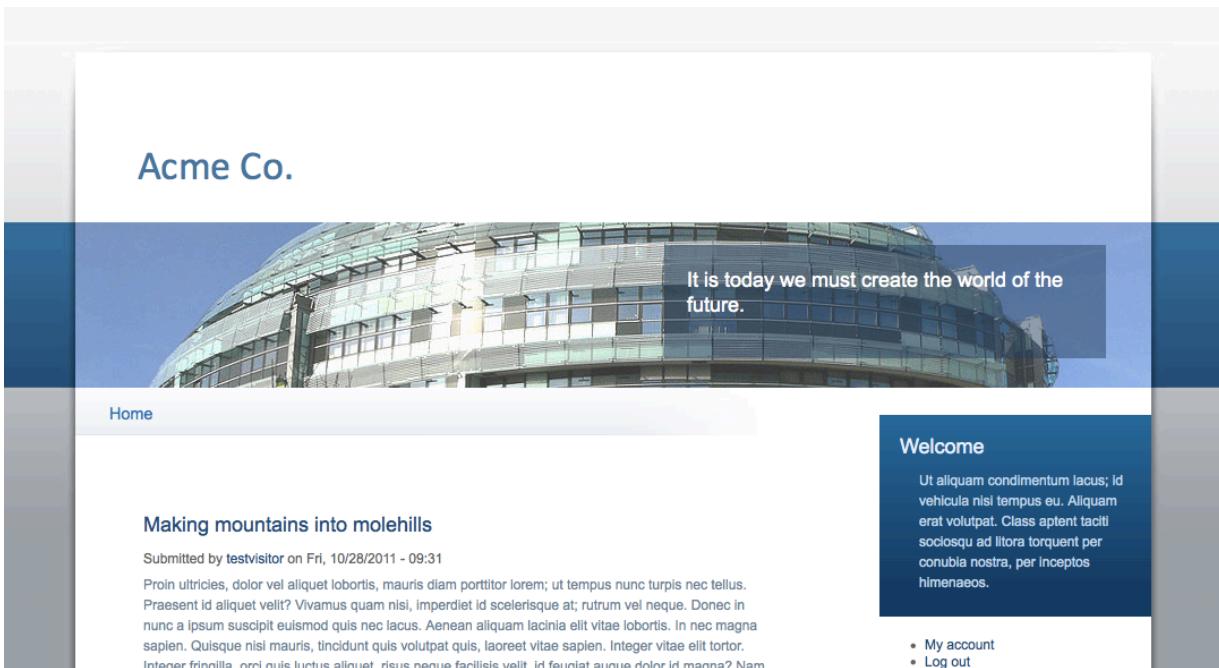
Seven (administration theme)

- None -

Bartik

- None -

3. Click **Save block**.
4. Check your site! You can edit your site title and slogan and add in some articles to see how the site will look with content.



Theme customization

How can you customize the banner images in the Busy theme? You simply drop the images into the theme folder. What would happen if your client would like to add images? They would need root access to the site? That is why there are configurable themes.

For the purposes of explanation we can roughly group theme development methods into two:

Static customizations

Often these themes have customizations done directly in the theme folders. For example, you may be able to override or add images and change the look and content of the site.

- **Blue Masters** <http://drupal.org/project/bluemasters>
- **Marinelli** <http://drupal.org/project/marinelli>

Theme configuration

Some of these themes are based on core theme systems such as Adaptive theme or Fusion. Or they work with options like the color module. They give you control within the UI for some levels of design. You can adapt using sub-themes to control what your clients can configure.

- **Adaptive theme** <http://drupal.org/project/adaptivetheme>
- **Fusion** <http://drupal.org/project/fusion>

- The screenshot below shows the Corolla <http://drupal.org/project/corolla> theme settings page, where you can control the positioning of sidebars or color, among other things.

The screenshot shows the 'Appearance' section of a Drupal site titled 'Acme Co.'. The top navigation bar includes 'LIST', 'UPDATE', and 'SETTINGS' buttons, along with tabs for 'Global settings', 'Bartik', 'Corolla' (which is selected), and 'Seven'. A breadcrumb trail indicates the path: Home > Administration > Appearance > Settings. Below the navigation, a note states: 'These options control the display settings for the *Corolla* theme. When your site is displayed using this theme, these settings will be used.' On the left, a sidebar lists several options: Standard Layout, Tablet Layout, Smartphone Layout, Fonts, Font Size, Heading Styles, and Rounded corners. The main content area is titled 'Standard Layout' and describes it as suitable for desktops, laptops, and large screen devices. It features a section titled 'Choose sidebar positions' with three radio button options. The first option shows a blue sidebar on the left and a yellow sidebar on the right. The second option shows a yellow sidebar on the left and a blue sidebar on the right, which is the selected configuration. The third option shows two blue sidebars. A cursor icon is visible on the right side of the interface.

Unit 1.4 - Make a simple theme

Let's see in detail what the minimum requirements are for a theme. Slowly we'll build up and develop this throughout the session.

Documentation: What's in the info file?

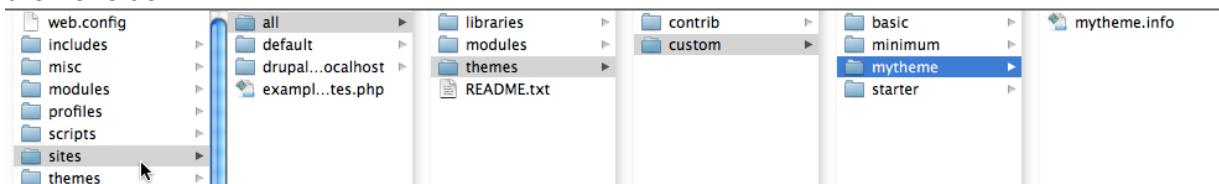
There are a number of declarations you should make in the info file. See the Drupal 7 Theme Guide for more information: <http://drupal.org/node/171205>

Exercise – Create your first theme

A. Create a theme directory and an .info file.

Your themes must be located in sites/all/themes. Optionally, you can organize *contrib* (contributed) and *custom* themes into their own folders.

1. Create the folder: "mytheme" in your sites/all/themes/**custom** folder. This is now your own theme folder.



2. Create a plain text file: mytheme.info in the mytheme folder.
3. Edit the mytheme.info file. Let's start with the basics:

```
name = My great theme
description = A great theme
core = 7.x
engine = phptemplate
```

4. Save!

B. Clear your cache!

Drupal 7 stores a cache of the data in your theme's .info file, you MUST refresh Drupal 7's cache by simply visiting the Appearance page.

The theme registry

<http://drupal.org/node/173880#theme-registry>

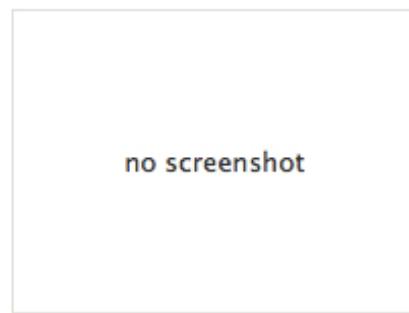
Drupal caches information about the available theme functions and templates in the theme registry. When adding a new template or function, the theme registry needs to be cleared. This is not needed when editing existing templates or functions, only when adding or removing.

C. Enable the theme in the Drupal admin and set it to default.

1. Go to Appearance and locate your new theme under the Disabled themes list.
2. Click **Enable and set as default**.

D. Review your new theme

1. Close the admin overlay and visit your site.
 - What can you tell from the look of your site about how Drupal works?
 - What was unexpected?
2. View source. What CSS files are loaded? Browsing around the site, you will notice there is some basic styling, for example to layout the view/edit tabs.



My great theme

A great theme

[Enable](#) | [Enable and set default](#)

About us

[View](#) [Edit](#)

3. Go to Structure > Blocks to see the Blocks administration page. Review what regions are available. Although your theme didn't declare regions, it took the **default regions**.

The screenshot shows the 'Blocks' administration page. On the left, there is a list of blocks: 'User menu', 'Who's new' (which is selected and highlighted in grey), and 'Who's online'. To the right, there is a dropdown menu for assigning regions. The dropdown menu has options: '- None -', 'Header', 'Highlighted' (which is selected and highlighted in blue), 'Help', 'Content', and 'Left sidebar'. A cursor arrow points to the 'Highlighted' option.

Exercise - Add a screenshot

We'll add a screenshot, comments and a description. We'll compare your theme with the "minimum theme" example and copy what we need.

1. Go to <https://github.com/acquia/training/tree/master/themes/>
2. Click the "mytheme" folder and download the screenshot.png to *your local* mytheme folder.
3. Edit your .info file, after the name and description, add the screenshot declaration in bold.

```
name = My theme
description = A great theme
screenshot = screenshot.png
```
4. Save, and refresh your Appearance page to see the changes.

Screenshots for Drupal themes

Screenshots should be sized to **294x219**. This is 30% of an original screenshot at ~960x720 (4:3 ratio). You can read about the screenshot guidelines if you want to contribute your theme to Drupal.org <http://drupal.org/node/647754>

Exercise – Add regions

A. Let's declare regions in our info file and see the differences.

1. Add the following lines to your mytheme.info file. *Comments start with a semi-colon.*

```
; Regions  
; -----
```

2. Add the following lines. Each region declaration is arbitrary. This means you can name regions to suit your needs. **The order you put them in here, is the order they appear in the block configuration screen.**

```
regions[highlighted] = Highlighted  
regions[content] = Content  
regions[sidebar_first] = Left sidebar  
regions[sidebar_second] = Right sidebar
```

3. Add the following lines. These are required core regions, but they are *hidden* and won't appear in the block region list. "If you define custom regions, it is important to remember that you need to include the page_top and page_bottom regions in your set of regions." See <http://drupal.org/node/254940#closure>

```
; Required core regions  
; -----  
regions[page_top] = Page top  
regions[page_bottom] = Page bottom
```

4. Save. And clear your cache by visiting the Appearance page.
5. Let's see what changed. Go to Structure > Blocks to review the block settings. Click **Demonstrate block regions**. Are the other defaults there?
6. Go to your front page. What happened to your block settings and positions?

As you see, **if you want to add additional regions, you must also declare the defaults.**

7. Edit your mytheme.info file and add the following custom regions **in bold**. *The order is arbitrary, you may like to put these in the order of appearance in the flow of the document.*

```
; Regions  
; -----  
regions[header] = Header  
regions[highlighted] = Highlighted  
regions[help] = Help  
regions[content] = Content  
regions[sidebar_first] = Left sidebar  
regions[sidebar_second] = Right sidebar  
regions[footer] = Footer
```

- After you make changes to the .info file, you must clear your cache.

A note about hidden regions

From documentation: <http://drupal.org/update/themes/6/7#closure>

When doing site-specific themes, it might also be useful to add more hidden regions (to provide ways for modules to add output to more places in the theme without defining blocks showing up on the blocks interface), you can do that via the regions_hidden[] .info file array which is new to Drupal 7:

```
regions[indicators] = Indicators
regions_hidden[] = indicators
```

B. Re-adjust block settings

So we messed up the block positions to prove a point. Now you can fix them.

- Go back to administer your site at Structure > Blocks to fix your block settings. Check if the appropriate blocks are in Help and Footer regions.
 - Footer:** Powered by Drupal
 - Help:** System help
- Adjust the following blocks as well:
 - Right sidebar:** Disable all blocks.
 - Header:** Move the Search form to the Header.
- Review your site. Do you see "Powered by Drupal" at the footer of your site now?

Exercise – Add features

"Features" can be configured at the Theme settings page. Go to your theme's settings page and see what options are available.

The screenshot shows the 'Appearance' settings page for the 'My great theme' theme. At the top, there are tabs for 'LIST' and 'SETTINGS'. Below the tabs, the breadcrumb navigation shows 'Home > Administration > Appearance > Settings'. There are also tabs for 'Global settings', 'Bartik', 'My great theme' (which is selected), and 'Seven'. A note below the tabs states: 'These options control the display settings for the *My great theme* theme. When your site is displayed using this theme, these settings will be used.' A large box titled 'TOGGLE DISPLAY' contains a list of checkboxes for various page elements. All checkboxes are checked:

- Logo
- Site name
- Site slogan
- User pictures in posts
- User pictures in comments
- User verification status in comments
- Shortcut icon
- Main menu
- Secondary menu

- Modify your theme's features. Add the following lines to your info file:
;
Features

```
; -----  
features[] = logo  
features[] = name
```

2. Save, clear cache and test your theme. When you return to the theme Settings page, what do you see?

3. Modify your info file and add the following lines:

```
; Features  
; -----  
features[] = logo  
features[] = name  
features[] = slogan  
features[] = node_user_picture  
features[] = comment_user_picture  
features[] = favicon  
features[] = main_menu  
features[] = secondary_menu
```

4. Save, clear cache and visit your theme Settings page. Now what do you see?



Unit 1.4 - Catch up!

If you think you've fallen behind, refer to the instructions for replacing your current "mytheme" folder with the "mytheme_1_4" folder. This will prepare you for the next unit.

Unit 1.5 - Add CSS and JavaScript

By adding your own custom CSS files you can override CSS declarations which come with core or contributed modules. You can also add CSS declarations.

Documentation: Core CSS

You can identify CSS to override through tools like Firebug. This documentation will also help.

- Standard Drupal core styles and classes <http://drupal.org/node/388372>
- Core Block CSS IDs <http://drupal.org/node/778884>
- Drupal CSS coding standards <http://drupal.org/node/302199>

Exercise – Step-by-step add a CSS file

Up until now you've been working with the bare bones of Drupal's theme system. It's CSS which makes the pages shine.

A. Prepare the site (optional)

This is optional, but will make your site appear similar to the screenshots.

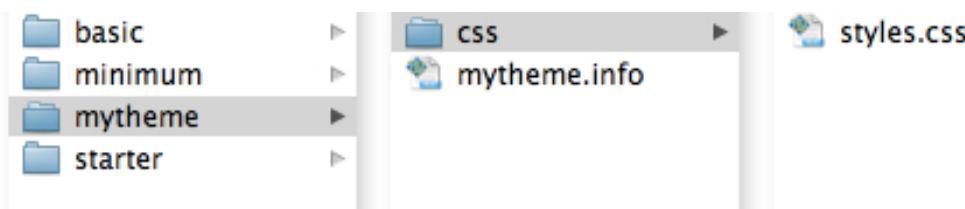
1. Modify your site title so it is "Global Travel". Add a slogan.
2. Add an article.

B. First, create the CSS file.

1. Create a "css" folder in your mytheme folder.
2. Add a styles.css file to the folder, use the sample CSS below to make a visible change to your site. (If you're new to CSS, pay close attention to opening and closing curly brackets.)

```
body {  
    font-family: sans-serif;  
    padding: 200px;  
}  
h1 {  
    background-color: #efefef;  
}  
#page {  
    width: 960px;  
    margin: 0 auto;  
}
```

3. You now have a new folder and CSS file like this:



C. Declare the style sheet in the info file

Next, add the style sheet declaration into the mytheme.info file. Use the comments to help keep your file organized.

```
; Stylesheets  
;  
-----  
stylesheets[all][] = css/styles.css
```

Adding a media property and media query CSS files

You can define the "media property" for a style sheet. For example:

```
; Add a style sheet for print media  
stylesheets[print][] = thePrintStyle.css
```

You can also do media queries. For example:

```
; Add a style sheet with media query  
stylesheets[screen and (max-width: 480px)][ ] = iPhone.css
```

Read more <http://drupal.org/node/171209>

D. Save and test

Remember, Drupal keeps a record of the data in the .info file. Another way to clear cache is to go to Configuration > Development > Performance and click **Clear all caches**. Also clear your browser cache! You can see we have some basic styling now.

The screenshot shows a Drupal website with the following structure:

- Header:** A blue logo icon (a stylized drop) and the text "Global Travel".
- Slogan:** "Our great slogan."
- Main menu:** "Main menu" with a sub-item "Home".
- Secondary menu:** "Secondary menu" with sub-items "My account" and "Log out".
- User profile:** A placeholder silhouette image of a person.
- Content area:** A section titled "This is an article" containing the text "Submitted by siteadmin on October 29, 2011". Below this is a long, mostly illegible paragraph of placeholder text (Lorem ipsum).

Exercise – Adding Javascript

1. Create an empty folder called "js" in our theme folder.
2. Create an empty text file called mytheme.js and save it in your theme's CSS folder. So you have sites/all/mytheme/js/mytheme.js
3. Javascript can be added with a similar declaration to CSS. Edit your .info file and add this line:
`scripts[] = js/mytheme.js`

Challenge Exercise: Install a third party CSS file, Normalize

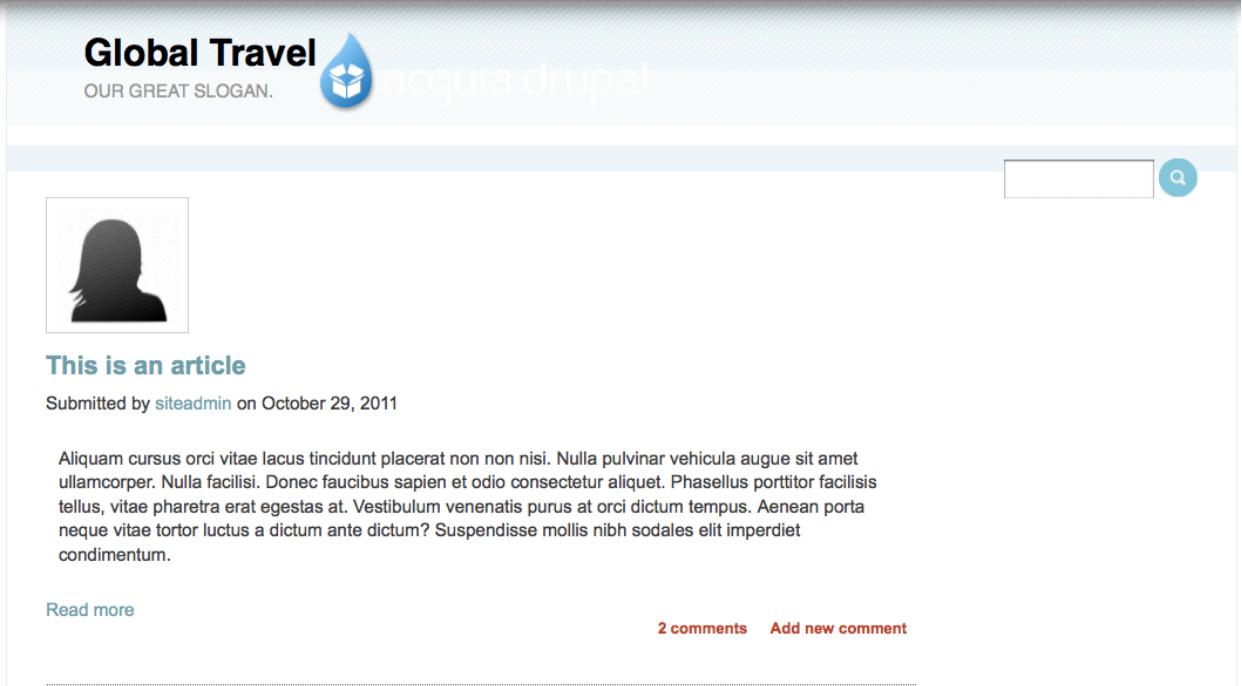
Using third-party CSS files like Normalize can save you countless hours of browser testing.

"Normalize.css is a customizable CSS file that makes browsers render all elements more consistently and in line with modern standards. We researched the differences between default browser styles in order to precisely target only the styles that need normalizing."

1. Get the normalize.css file at <http://necolas.github.com/normalize.css/>
2. Save to your "mytheme" CSS folder, so you have sites/all/mytheme/css/normalize.css
3. Follow the example for adding a CSS declaration to your .info file.
4. TIP: Add the third-party CSS file in the line before the styles.css file otherwise some of your declarations may be overridden.
5. Clear your cache, and check your site.

Exercise – Practice adding more CSS files

At this point you can get quite far in styling your site by altering the CSS alone. We'll start customizing the look of our site now. We've added extra content to demonstrate how the site will look. For example, a site title and an article with two comments.



The screenshot shows a Drupal website with the following components:

- Header:** The site has a header with the title "Global Travel" and a subtext "OUR GREAT SLOGAN". It also features the Acquia Drupal logo.
- User Profile:** A user profile picture of a woman is displayed on the left side of the page.
- Search Bar:** A search bar with a magnifying glass icon is located in the top right corner.
- Article Content:** An article titled "This is an article" is shown. It was submitted by "siteadmin" on October 29, 2011. The content of the article is a long block of placeholder text (Lorem ipsum).
- Article Footer:** Below the article content, there are links for "Read more", "2 comments", and "Add new comment".

A. Add images to your mytheme folder

1. Create a folder called "images" in your theme folder.
2. Go to <https://github.com/acquia/training/tree/master/themes/mytheme> and locate the **images** folder. Download these image files into your theme's images folder:
 - bg-header.png
 - search.png
 - bg-block-title.png
3. If you have time you can add the remaining images from github to your theme's images folder.

B. Add the CSS files and declaration in the .info file

1. Locate three style sheets in the Github mytheme CSS folder, and copy and save them to your site's mytheme CSS folder:
 - html-elements.css
 - layout.css
 - styles.css (you can use this to replace your current styles.css file)
2. Use the procedure in "Step-by-step Add CSS" exercise to add these new CSS files to your info file.
3. Clear cache and refresh.



Unit 1.5 - Catch up!

If you think you've fallen behind, refer to the instructions for replacing your current "mytheme" folder with the "mytheme_1_5" folder. This will prepare you for the next unit.

Unit 1.6 - Anatomy of a theme

Your instructor will do a presentation about the anatomy of the theme system. You can see how templates are nested and called. Refer to Drupal.org Documentation

- Core templates <http://drupal.org/node/190815>
- Overview of theme files <http://drupal.org/node/171194>

Typical theme contents

.info file

- A .info file is the only required file of a theme. It defines the theme name, its regions, stylesheets, and other options. Each line defines an option as a key-value pair (key = value). Values not specified will use Drupal's default.
- **Info file contents:** See <http://drupal.org/node/542202>

core and engine (required) compatibility information.

regions(optional) The regions in which blocks can be placed

stylesheets(optional) Add additional stylesheets to the theme

features (optional) Page elements to be toggled via configuration

template.php

- Additional processing and logic for a theme takes place here. It can override output of other modules via theme function overrides. See <http://drupal.org/node/173880#function-override>
- It can define and edit variables to be used by template files with preprocess functions
See <http://drupal.org/node/223430>

template files

- These files end in *.tpl.php. Default templates include page.tpl.php, node.tpl.php, block.tpl.php and others. See <http://drupal.org/node/1089656>
- The theme registry stores theme information, it must be cleared when new files are added.
- See <http://drupal.org/node/173880#theme-registry>
- Use folders to organize your template files

css files

- Files need to be listed in the .info file or included elsewhere with code.
- Multiple files can be created to keep things organized.
- Use folders to organize your CSS files

js files

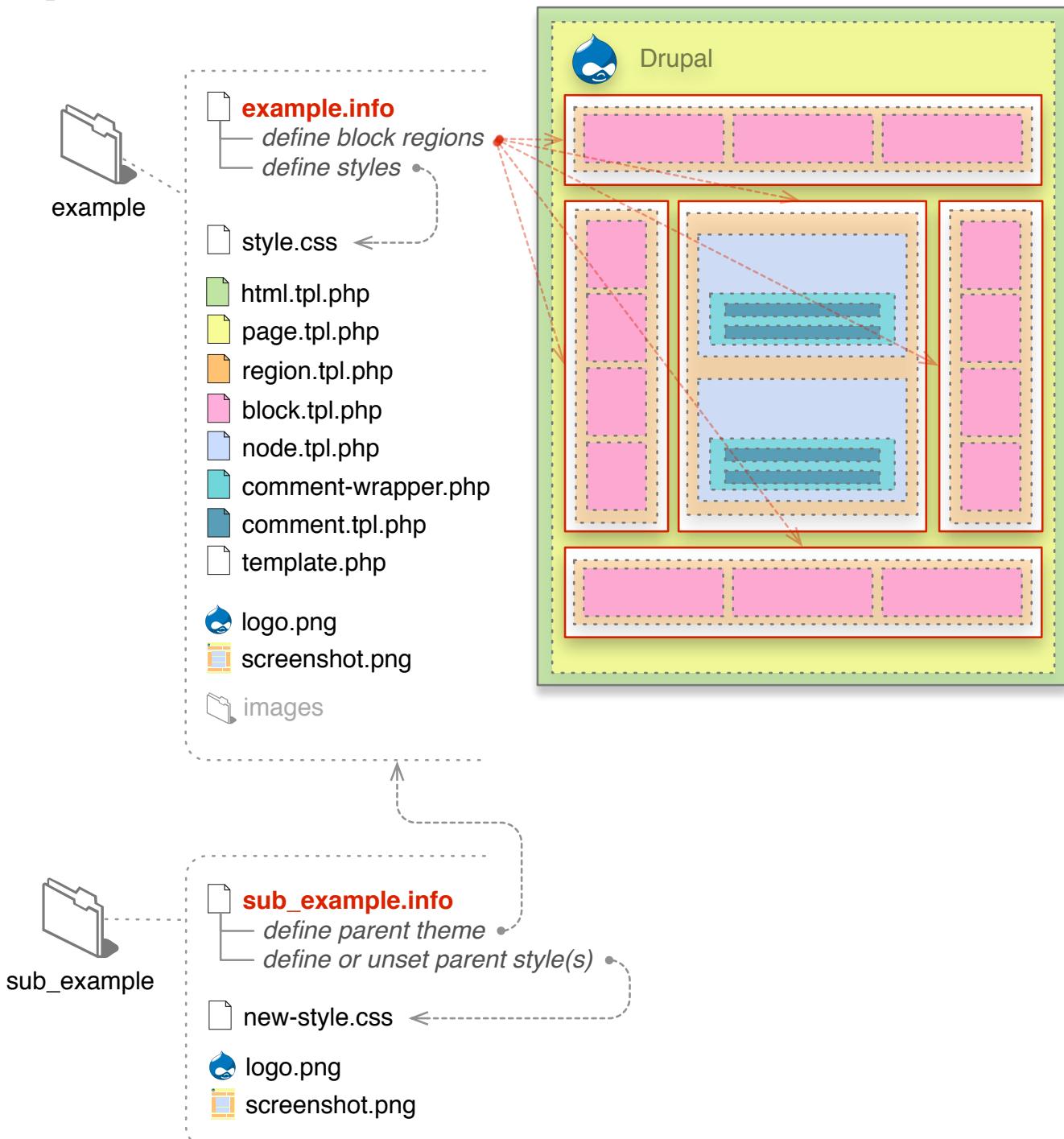
- These must be included from code, typically from inside template.php

images

- Image files that are specific to the theme (background images, buttons, etc) should be kept within the theme in an images folder
- Images are mainly called by CSS files as backgrounds.

- Images that are not specific to the theme (photo placed inline in a node body, etc) should not be placed in the theme, but should be handled by Drupal and placed in its files directory

Drupal theme construction



Unit 1.7 - Resetting your site

Before we go further in the site, you might like to capture a snapshot, and also reset your theme so it matches the one in the manual

A. Save a snapshot

Follow the instructions at the start of the document on how to save a snapshot.

B. Theme files catch up

If you think you've fallen behind, refer to the instructions for replacing your current "mytheme" with the "mytheme_2_0". This will prepare you for the next session.

Session review

In this Session, you looked at contributed themes in-depth. We created a theme from scratch to see how the Drupal theme system works.

- You also set up and started customizing your own theme.
- You saw how to add new template files, connect CSS and JavaScript files.
- You started customizing it by hand-writing code, by copying and pasting code, and by copying and modifying code from examples.
- Step by step, you got a quick overview of the main elements of a typical theme, even preprocess functions which we will explore in more depth soon.
- You have three working examples of three custom themes: Minimum, Basic and Starter theme.

Next we'll look at the theme layer in more detail.

Session 2: Customize with templates

In this session you'll learn some typical areas designers tend to forget to style and how you can avoid these pitfalls. You'll learn how to identify where Drupal is generating specific markup.

Unit 2.1 - Tools for theme developers

Your instructor will do a presentation and demonstration of using tools to inspect your site.

- How to find out 'What in Drupal is controlling that?'
- Firebug. Identify CSS declarations and in which files.
- Theme Developer module. Locate current template and template suggestions.

Exercise – Download and install Devel and Theme Developer

- Install the "Theme Developer" module available at http://drupal.org/project/devel_themer
- This also requires the "Developer" module available at <http://drupal.org/project/devel>

Using the Theme Developer

After you enable this module, you'll get a small check box in the lower corner of your site. Enable this option and click to inspect any element on your page. It will tell you which template is generating that output. There are "candidate templates" for more specific options.

The screenshot shows a Drupal page with a user profile picture and the title "This is an article". Below the title is the date "October 29, 2011" and the author "siteadmin". A tooltip from the "Themer info" module is displayed over the page content. The tooltip has a dark background and white text. It shows the template hierarchy: "Parents: block < region < page < html". The main template called is "sites/all/themes/custom/mytheme/templates/node--article.tpl.php", which is highlighted with an orange border. Below it, "Candidate template files:" are listed: "node_2 < node_article < sites/all/themes/custom/mytheme/templates/node--article.tpl.php". The "Preprocess functions:" listed are "template_preprocess + template_preprocess_node + contextual preprocess + rdf_process_node + mytheme preprocess_node". The "Process functions:" listed are "template_process + rdf_process". The "Duration: 17.87 ms" is shown. At the bottom of the tooltip, there is a box containing "(Array, 62 elements)".

Themer info

Under development: Watch file names

This module is currently under development. At the time of writing there is a bug that offers example candidate files with underscores __ where they should be double dashes --



Challenge exercises:

- Add new CSS declarations to your theme.
- Add another template file based on a suggestion.

Next, we'll practice what we've learned so far in identifying points of interception and overriding and practice template-based overrides.

Unit 2.2 - Introduction to overriding template files

Template files contain more HTML and should have minimal logic (PHP). We will work on template file overrides in more detail. At this stage, familiarize yourself with the terminology and how it works.

Documentation: About core templates and template suggestions

Keep these documentation pages bookmarked. They are your theming friends.

- A list of the core templates These are the default template (.tpl.php) files provided by core in Drupal 7 <http://drupal.org/node/190815>
- Drupal 7 Template Suggestions <http://drupal.org/node/1089656>

Exercise – Override a core template - node.tpl.php

In these exercises you *copy* the core template file into your own theme folder to override.

A. First, some facts about templates:

- Templates end with .tpl.php
- Drupal select templates from the most specific to the most general.
- Your theme templates override the core templates.
- Your theme can override existing templates included with core or a module by simply having a file with the same name.
- Multiple template files are used for each page load.
- They are rendered from the inside out.
field.tpl.php > node.tpl.php > region.tpl.php > page.tpl.php > html.tpl.php

B. Override the core node.tpl.php file.

1. In the mytheme folder, make a new "templates" folder to keep template files organized.
2. Locate the core node template file in your site's root folder: /modules/node/**node.tpl.php**
3. **Copy** and save the **node.tpl.php** template to your theme folder. It should now be in sites/all/themes/custom/mytheme/**templates**/node.tpl.php.
4. Open the file. Make a visible change. For example add this line:
`<h1>I am a node!</h1>`
5. Save it.
6. Clear theme cache. Can you see how the node template was overridden?
7. Delete the alteration you made to the template file. Save. (No need to clear cache).

Exercise – Override node templates by node ID and content type.

A. First, some facts about specific template suggestions.

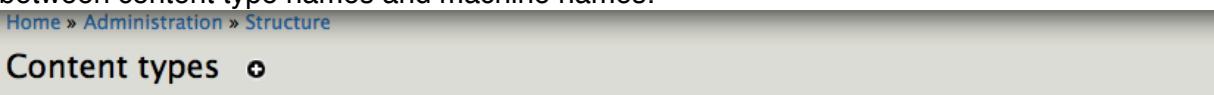


page.tpl.php



page--front.tpl.php

- Drupal will use the most specific template it finds. Read more at <http://drupal.org/node/1089656#page-suggestion>
- More specific templates can be created based on a node type, a specific page, etc. These are alternatives to the existing templates included with core or a module. You can use the patterns called "template suggestions" to do this.
- Template suggestions use the **machine names**. For example, you can see the difference between content type names and machine names:



The screenshot shows the 'Content types' section of the Drupal administration interface. It includes a header with 'Home > Administration > Structure' and a sub-header 'Content types'. Below is a table with two rows:

NAME	OPERATIONS
Article (Machine name: article) Use <i>articles</i> for time-sensitive content like news, press releases or blog posts.	edit manage fields manage display delete
Basic page (Machine name: page) Use <i>basic pages</i> for your static content, such as an 'About us' page.	edit manage fields manage display delete



The screenshot shows the 'Content types' section of the Drupal administration interface. It includes a header with 'Home > Administration > Structure' and a sub-header 'Content types'. Below is a table with two rows:

NAME	OPERATIONS
Article (Machine name: article) Use <i>articles</i> for time-sensitive content like news, press releases or blog posts.	edit manage fields manage display delete
Basic page (Machine name: page) Use <i>basic pages</i> for your static content, such as an 'About us' page.	edit manage fields manage display delete

- As you improve your coding skills with Drupal you can make your own template suggestions.
- Look at these example custom templates, what do you think they affect?

based on	template pattern	example custom template
node.tpl.php	node-- type .tpl.php	node--article.tpl.php
node.tpl.php	node-- nodeid .tpl.php	node--15.tpl.php

based on	template pattern	example custom template
block.tpl.php	block--region--delta.tpl.php	block--sidebar--first.tpl.php

B. Make a template by node type - article

In this case we're going to change the display of articles. Based on the steps from node template exercise, you will now override the node template for articles.

1. First, if you haven't already, go to Add content > Article and make a new sample article so we can see the changes.
2. Determine the name of your new file based on the patterns described above.
3. Duplicate the node templat, giving it the correct name.
4. Make a visible alteration such as adding this line.
`<h1>I am an article!</h1>`
5. Save the changes and clear your cache.
6. Go to view your article. Do you see your message appear?

Unit 2.3 - PHP basics: Conditional statements

Conditional statements allow some basic logic: If X then Y. This lets you test for conditions being met, and allows you to control how the templates are rendered.

1. Reading code is easier when you can see what it says in English. Take this as an example:

```
if($page['front'])  
{  
    print ("This is the front page");  
}
```

This says, if this is the front page, then print this message. Read more about If statements on <http://php.net/manual/en/control-structures.if.php>

2. However, escaping the various attributes in HTML can be tricky in a print statement. So there's a better syntax which opens the conditional statement and closes it after a chunk of HTML.

```
<?php if ($page['front']): ?>  
Print some HTML  
<?php endif; ?>
```

3. Find out more

- Comparison operators <http://php.net/manual/en/language.operators.comparison.php>
- Logical operator <http://www.php.net/manual/en/language.operators.logical.php>
- About alternative syntax <http://www.php.net/manual/en/control-structures.alternative-syntax.php>

Unit 2.4 - Practice extending your theme

In Drupal 7 you have two options to control the appearance of the main menu. By default you can toggle the appearance in the

Now, we'll add in a new region. We can assign blocks to it, but they won't show up until we print them to the page.tpl.php file.

Exercise – Override a core template and add a region - page.tpl.php

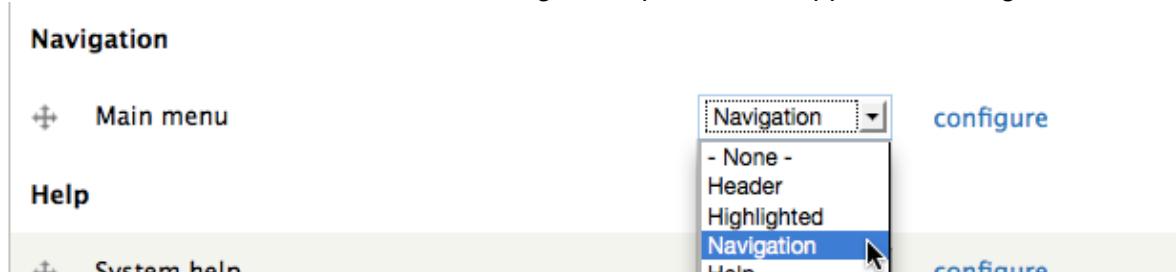
In this exercise we'll edit the info file, and add a template as a final review of what you've learned so far.

A. Add the region in the .info file.

1. Edit your mytheme.info file and add the following bolded region, "Navigation".

```
; Regions  
;  
regions[header] = Header  
regions[navigation] = Navigation  
regions[highlighted] = Highlighted  
regions[help] = Help  
regions[content] = Content  
regions[sidebar_first] = Left sidebar  
regions[sidebar_second] = Right sidebar
```

2. Save the info file. Go to Appearance to reload the .info file settings.
3. Go to Structure > Blocks. Locate the *Navigation* option which appears as a region now.



B. Prepare the site

In order to see this working, we have to prepare the site.

1. Go to Structure > Blocks and place the Main menu in the *Navigation* region.
2. Configure the Main menu block to remove the block name by setting the block title to <none>.

3. Go to Appearance > Settings > My theme

Home > Administration > Appearance > Settings

Appearance ⊕

LIST **UPDATE** **SETTINGS**

Global settings Bartik My theme Seven

These options control the display settings for the *My theme* theme. When your site is displayed using this theme, these settings will be used.

TOGGLE DISPLAY

Enable or disable the display of certain page elements.

Shortcut icon

Main menu

Secondary menu

4. De-select both the Main menu and Secondary menu options.

5. *Optional: You should remove these "features" from your theme's .info file. Can you think of why?*
6. Click **Save configuration**.

C. Override the page.tpl.php

1. Go to your root directory. Locate **modules/system/page.tpl.php**
2. Copy and save this template to your mytheme folder. It should now be in sites/all/themes/custom/mytheme/templates/page.tpl.php
3. Open the page.tpl.php file from your theme's folder. Locate lines 104-109 which print out the main and secondary menus. This screenshot shows the code collapsed so you can see the starting and ending lines.

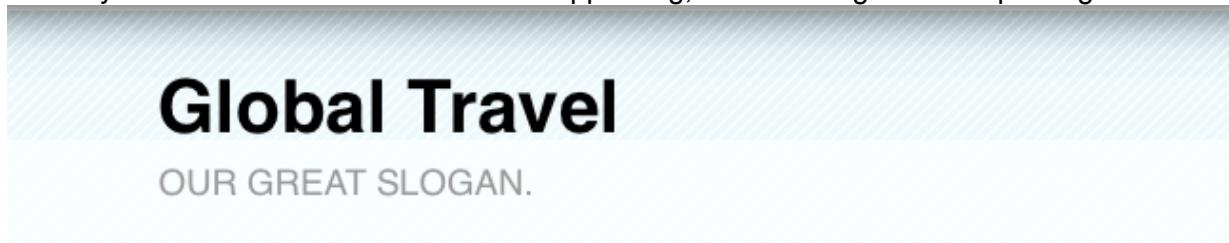
```
104  <?php if ($main_menu || $secondary_menu): ?>
105  |   |<div id="navigation"></div> <!-- /.section, #navigation -->
109  |   |<?php endif; ?>
```

4. Replace with the following lines:

```
<?php if ($page['navigation']): ?>
<div id="navigation"><div class="section">
  <?php print render($page['navigation']); ?>
</div></div> <!-- /.section, #navigation -->
<?php endif; ?>
```

5. Save your new page.tpl.php file. And, clear your cache.

6. Check your site now. The correct menu is appearing, *but* the design needs updating.



- [About](#)
- [Articles](#)
- [Home](#)

D. Add CSS for the Navigation region

You may have added the navigation.css in an earlier "challenge" exercise.

1. Access the navigation.css file from the "mytheme" folder on <https://github.com/acquia/training/tree/master/themes/>
2. You have 2 options.
 - Edit the styles.css file and paste in the declarations from navigation.css file directly.
 - Or, use the steps in "Add CSS" exercise to add the navigation.css to your .info file.
3. Save the file. Clear cache and your browser cache.
4. View your site! Yours may look different. Add some menu items to see how it will look.



E. Discussion: compare approaches

The default core theme hard-codes the main and secondary menus into page.tpl.php file. Menus however create blocks which you can move and place throughout the theme. We changed it so you configure the menu to appear in the new Navigation region.

What would be the advantages or disadvantages of either approach?

Challenge Exercise – Override the maintenance mode page.

What is maintenance mode?

1. Go to Configuration > Development > Maintenance mode.
2. Select "Put your site into maintenance mode".
3. Save configuration.

4. Look at your site in another browser, or logout and look at your site. What differences do you see?

Change the look of this page

Using the steps from node template, override the "maintenance mode" template.

1. Copy the template into your theme folder. The core template file is located in your root folder: modules/system/maintenance-page.tpl.php
2. Make a visible alteration and save the changes.
3. Clear cache and view your site logged out. Does your maintenance mode page look different?
4. Log back in. Go to your site- <http://example.com/user> for the login page.



Unit 2.4 - Catch up!

If you think you've fallen behind, refer to the instructions for replacing your current "mytheme" folder with the "mytheme_2_4" folder

Unit 2.5 - Theming contributed modules

Explore theming contributed modules

- Download and install the Weather module <http://drupal.org/project/weather>. This is a simple module which pulls in real-time data from another source. It's a good example of how a custom or contributed module should be made to allow the themer to modify markup.

1. Enable the Weather module. Open the Install.txt file from the Weather module.
2. Go to Configuration > User Interface > Weather.
 - a. Click to **+ Add a display** and follow on-screen instructions to adjust settings. Save the display. You will see this listed and noted as System-wide display (number 1).
 - b. Click to Add a location. Follow on screen instructions to add your favorite location.

Home » Administration » Configuration » User interface

Weather 

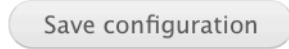
You can add, edit, and delete locations from system-wide weather displays. Moreover, you can specify default values for newly created displays.

a.  **+ Add display**  **+ Edit default display**

SYSTEM-WIDE DISPLAY (NUMBER 1)	WEIGHT
b.  Add location to this display	

Directory for custom images

Override the default image directory. This directory must be a subdirectory of the Drupal 'files' path.

 **Save configuration**

3. Go to Structure > Blocks and place the new block "Weather: system-wide display (number 1)" in the **right sidebar**.
4. Turn on the Theme Developer, and return to the home page.

5. Inspect the block using the Themer info tool. Determine what template is rendering this block. How many templates are being used?

The screenshot shows a weather block titled "Current weather" for the location "Langkawi". The block displays a night sky icon with clouds and a crescent moon. Below the icon is a list of weather details:

- Broken clouds
- Temperature: 26 °C
- Wind: Northeast, 3.7 km/h
- Pressure: 1010 hPa
- Rel. Humidity: 94 %
- Visibility: 10 km

To the right of the block is a "Drupal Themer Information" panel. It shows the following details:

- Parents: `weather_theming` < `block` < `region` < `page` < `html`
- Template called: `sites/all/modules/weather/weather.tpl.php`
- Candidate template files: `sites/all/modules/weather/weather.tpl.php`
- Preprocess functions: `template_preprocess` + `contextual preprocess`
- Process functions: `template_process` + `rdf_process`
- Duration: 0.73 ms
- ... (Array, 20 elements)

A "Themer info" button is visible at the bottom left of the main content area.

Override the Weather modules template

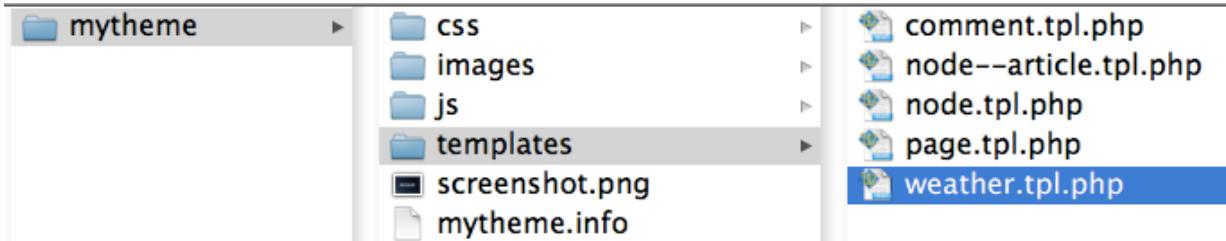
You can customize much of the look of this block, through CSS. However, you can get more control in the markup.

1. Open the weather module's folder.

The screenshot shows the contents of the "weather" module directory. The files listed are:

- weather.tpl.php
- weather.forms.inc
- weather_theme.inc
- weather_parser.inc
- weather_compact.tpl.php
- weather.info
- weather_data.txt
- README.txt
- LICENSE.txt
- INSTALL.txt
- CHANGELOG.txt
- tests
- images
- weather.module
- weather.install

2. Copy the weather.tpl.php file to your theme's template folder, like this:



3. Edit the weather.tpl.php file and make a visible change to the file. Save it!
4. Clear your theme cache.
5. Check your site and inspect with the Theme Developer module.
6. Which template is being loaded?
7. Open another one of your favorite contributed modules. What .tpl.php files are included? If it's a module that provides markup on the screen, or creates blocks, you'll find there are template files.



Unit 2.5 - Catch up!

If you think you've fallen behind, refer to the instructions for replacing your current "mytheme" folder with the "mytheme_2_5" folder

Unit 2.6 - Views template suggestions

Custom themes are usually created for a specific site and can be tightly tied into the Views, Field and other configurations of the site.

Overriding Views templates

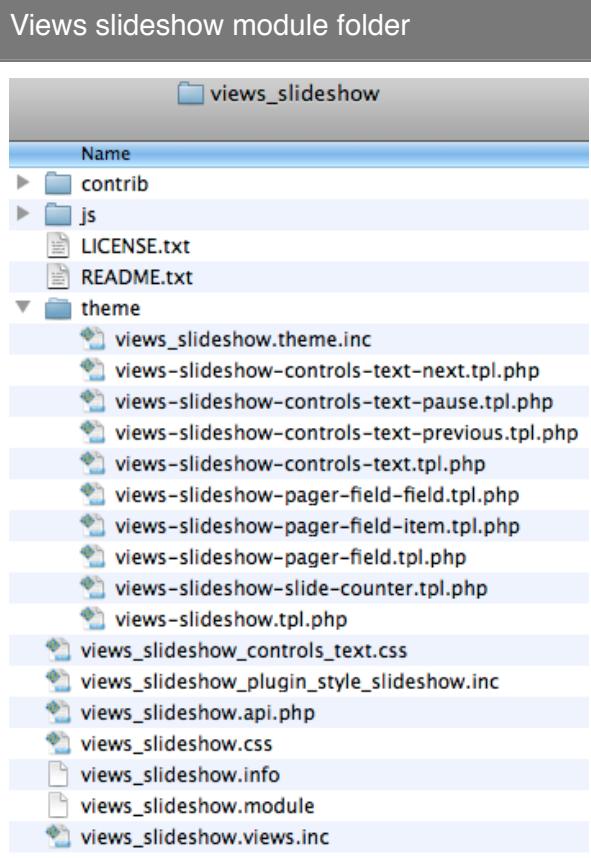
- Views templates are very similar to core templates, and have their own naming conventions based on their target. Templates can be specific to a given view, a particular field, an entire display style, or any combination of those.
- There is a hierarchy of templates which can be used to apply from either general to specific control of the output of Views.

Warnings and caveats!

- **Make sure to disable the Theme Developer module when working with Views.**
- **Be careful to not overload your template folder with many Views templates.** If you find you are using many Views template overrides, check what you are doing with your views. Control as much as you can through display and format configuration.

Contributed modules with Views templates

Some modules, such as Views slideshow offer templates for customization.

Views slideshow module folder	views-slideshow-slide-counter.tpl.php
	<p>Clip from the template:</p> <pre><div id="views_slideshow_slide_col class=<?php print \$classes; ?>"> 1 <?php print t('of'); ?> <?php print count(\$rows); ?> </div></pre>

Exercise – Override a views template

A. Prepare your site

Make sure you have the comments module enabled and some comments on the site.

B. Enable the Recent Comments view

1. If you don't already have Views installed and enabled, install it now. <http://drupal.org/project/views>
2. Go to Structure > Views. Local Recent Comments and click **enable**.
3. The **Recent comments** example view creates both a block and a page displaying the most recent comments with a link to the node. This allows you more control than the core "recent comments block".
4. Edit to see the preview. And click View page.

C. Explore the possible templates

1. Go to edit the recent comments view. Click to expand the "Advanced" column.
2. Click on Theme Information at the bottom of that column. This opens a pop-up window.
3. Here you see a list of possible templates from most general to most specific. Notice that there are different templates for display, style, row style and fields. This interface is somewhat like Theme Developer module, as it is showing you which template files are being used, the order

of the override possibilities, and other naming options.

Block: Theming information

My theme

- **Display output:** `views-view.tpl.php`, `views-view--comments-recent.tpl.php`, `views-view--default.tpl.php`, `views-view--default.tpl.php`, `views-view--block.tpl.php`, `views-view--comments-recent--block.tpl.php`
- **Style output:** `views-view-list.tpl.php`, `views-view-list--comments-recent.tpl.php`, `views-view-list--default.tpl.php`, `views-view-list--default.tpl.php`, `views-view-list--block.tpl.php`, `views-view-list--comments-recent--block.tpl.php`
- **Row style output:** `views-view-fields.tpl.php`, `views-view-fields--comments-recent.tpl.php`, `views-view-fields--default.tpl.php`, `views-view-fields--default.tpl.php`, `views-view-fields--block.tpl.php`, `views-view-fields--comments-recent--block.tpl.php`
- **Field Comment: Title (ID: subject):** `views-view-field.tpl.php`, `views-view-field--subject.tpl.php`, `views-view-field--comments-recent--subject.tpl.php`, `views-view-field--block.tpl.php`, `views-view-field--block--subject.tpl.php`, `views-view-field--comments-recent--block.tpl.php`, `views-view-field--comments-recent--block--subject.tpl.php`
- **Field Comment: Updated date (ID: timestamp):** `views-view-field.tpl.php`, `views-view-field--timestamp.tpl.php`, `views-view-field--comments-recent.tpl.php`, `views-view-field--comments-recent--timestamp.tpl.php`, `views-view-field--block.tpl.php`, `views-view-field--block--timestamp.tpl.php`, `views-view-field--comments-recent--block.tpl.php`, `views-view-field--comments-recent--block--timestamp.tpl.php`

Important! When adding, removing, or renaming template files, it is necessary to make Drupal aware of the changes by making it rescan the files on your system. By clicking this button you clear Drupal's theme registry and thereby trigger this rescanning process. The highlighted templates above will then reflect the new state of your system.

D. Copy `views-view-field.tpl.php` to your theme

1. Go to your root folder: `sites/all/modules/views/theme` folder and locate `views-view-field.tpl.php`
2. Copy this file to your theme folder so you now have `sites/all/themes/custom/mytheme/templates/views-view-field.tpl.php`

E. Rename the template file to apply to only the subject in the block display

Any changes made to `views-view-field.tpl.php` will affect every field in every view on your site.

Rename your theme's copy of the `views-view-field.tpl.php` to `views-view-field--comments-recent--block--subject.tpl.php`.

This targets just the Recent Comments view, only when displayed in a block.

F. Rescan template files

When adding a new template, as we've seen, you need to clear the theme registry. In the Theming information pop-up window, scroll down to click "Rescan template files" button.

- **Field Comment: Title (ID: subject)**: views--view-field.tpl.php, views--view-field--subject.tpl.php, views--view-field--comments-recent.tpl.php, views--view-field--comments-recent--subject.tpl.php, views--view-field--block.tpl.php, views--view-field--block--subject.tpl.php, views--view-field--comments-recent--block.tpl.php, views--view-field--comments-recent--block--subject.tpl.php
- **Field Comment: Updated date (ID: timestamp)**: File found in folder sites/all/themes/custom/mytheme/templates/

You will see your new template file in bold. Scroll over to see the location.

G. Make a visible change.

1. Find the following line:

```
<?php print $output; ?>
```

2. Make a visible change:

```
<div style="border:1px solid #ff0099;">
<?php print $output; ?>
</div>
```

H. Test! Save your files, clear your cache.

Go to Structure > Blocks. Locate the block View: Recent comments and place in your sidebar.

Now you should see the changes being loaded into your Recent comments.

Cleaning up Views Mark-up

Before considering adding template files, make sure you have configured Views for your optimum output. As we have seen, Views adds a wealth of classes in predictable patterns. This example shows a View with a HTML list. The View name is "content list". You can customize and clean up Views output.



A. A note about adding custom wrapper and element classes

- Locate and edit your Recent Comments View. In this example, we're working with a HTML list.
- Under Format, next to HTML List, click **Settings** to open the Style options.
- For example, you can customize a Wrapper class and List class on the page display.

- No changes are necessary at this time. Cancel and go back to the main Views UI.

Page: Style options

For All displays

Grouping field
- None -

You may optionally specify a field by which to group the records. Leave blank to not group.

Row class

The class to provide on each row. You may use field tokens from as per the "Replacement patterns" used in "Rewrite the output of this field" for all fields.

List type
 Unordered list
 Ordered list

Wrapper class
custom

The class to provide on the wrapper, outside the list.

List class

The class to provide on the list element itself.

Buttons: Apply (all displays) | Cancel

B. Clean up Row style classes

- Under Format in the first column, click to edit Show: Fields | **Settings**
- De-select *Provide default field wrapper elements*. Note you can make fields flow inline here.
- Click **Apply...**

C. Clean up field classes

- Click to edit each field. Scroll down to Style settings (click to expand).
- Deselect *Add default classes*.

STYLE SETTINGS

Customize field HTML

Customize label HTML

Customize field and label wrapper HTML

Add default classes

Use default Views classes to identify the field, field label and field content.

- Repeat for each field.
- You can also opt to customize the HTML for labels and fields and change the element to h1, div, etc.

Session summary

In this session you worked on a variety of exercises to get fine-grained control over the markup of your site. You worked mainly in the theme layer, but you can also see how a certain amount of configuration is needed. We'll explore that in more depth next.

You have seen how to do:

- A core template override
- Create a new template based on a suggestion.
- Override templates included with contributed modules.
- Create Views based templates, and field specific templates.

Compare and discuss:

The last exercise was a demonstration of overriding views output with templates. Here you saw the control you can get through the Views UI in customizing the output. What are the applications of either approach?

Challenge exercises:

These are some additional ideas to try.

- Override a contributed module template.
- Create a Views template from the recommendations.
- Modify HTML around the PHP variables.
- Edit the node.tpl.php to display tags before the body and display in a floating div.

Session 3: Theme development process

Some things are difficult to do with Drupal. At some point you will come across visual designs that go against the grain of Drupal. Your entire approach, based on another system may also color how you approach Drupal and make theme development more difficult than it needs to be. Let's look at how we'd approach a project and think like Drupal.

Unit 3.1 - Comparison of processes

How do you plan your sites?

Share a bit about your own process and how you think theming will fit into it. The discussion we have here today will be similar to the kinds of work you'll need to do to theme your own site.

Use a "content out" instead of "canvas-in" approach

Mark Boulton, leader in designing grid systems for the web promotes a content out approach. The variety of devices and screen sizes indicates that a canvas-in approach isn't sustainable.

Read more: <http://www.markboulton.co.uk/journal/comments/a-richer-canvas>

A Drupal process

A good case study "Wellstone Action: A Drupal Process Case Study" by Gorton Studios, 2009 demonstrates that Theming and final design are finalized at a later stage in development.

Wireframes, designs and prototypes have been developed in iterations throughout, however the final design and actual theming aren't done until later.

Continue reading at <http://drupal.org/node/413910>

Sustainable themes for rapid prototyping

Drupal is ideal for rapid prototyping. New functionality and content can be changed and altered quickly without coding. This is why it's better to let Drupal generate the necessary markup, rather than editing code. This will make your themes less brittle and more sustainable.

Read more: by Laura Scott <http://rarepattern.com/nodes/2011/theming-firehose-nb-designers-front-end-developers-new-drupal>

Unit 3.2 - Identify what we forgot to theme

Depending on the make up of your team you may be the visual designer, or you may need to communicate with a visual designer. Your instructor will give a presentation to highlight some typical pitfalls in preparing a design for Drupal.

What makes sites look "Drupally"?

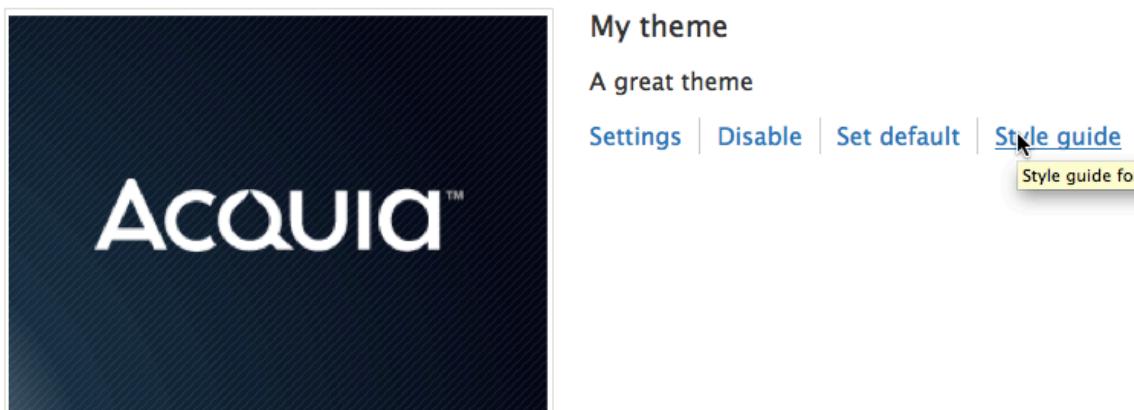
- Use the Style Guide module (described below) to inspect the Stark theme. This shows you what default Drupal styles are included.
- Refer to Nica Lorber's Fireworks file
http://www.chapterthree.com/blog/nica_lorber/design_drupal_template_approach

Identify "things we forgot to theme"

- Your Instructor will give a demo.
 - User login block and /user page. All account pages.
 - Editing screens (if not using Seven theme for content administration).
 - Drupal specific markup: Tabs, System messages and warnings.
 - Search results
 - Alternative media print styles
- If you allow your users to employ the WYSIWYG module and add markup to the site, make sure to account for that markup as well.
 - Block level elements: Headings (h1, h2, etc), ul, ol, blockquotes
 - Inline elements: Images (when added to text field).

Exercise – Download and install Style Guide module

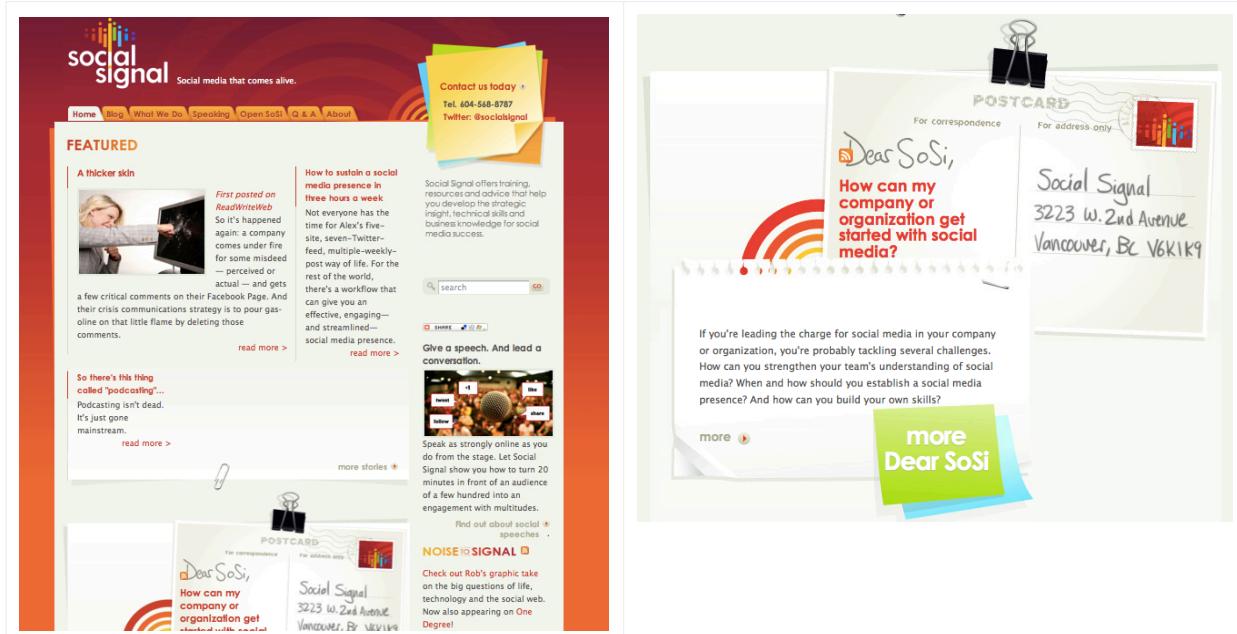
Install the "Style Guide" module available at: <http://drupal.org/project/styleguide>



After you enable the module, you will see a Style guide showing all the markup of your theme and how it is styled. <http://example.com/admin/appearance/styleguide>

Unit 3.3 - Case study: Social signal

We looked at this <http://socialsignal.com/> earlier, now we'll look at the site in more detail. Open the homepage and the Q & A page for comparison. We'll use paper or a white board to wireframe this site, and break down specific elements of the site.



How would you plan this site?

Imagine you received these images as a visual design composition.

- Identify how many regions the theme would need.
- How many blocks are on front page of the site?
- What content regions are consistent throughout the site?
- What contributed modules would you use?
- Is there a base theme or theme system you might choose to start with?
- Identify where you might need to do the following:
 - Override a template file
 - Override a string

Session summary

In this session we put theming practice in perspective. This will help prepare you for your next project. Before you do major changes to your theme, check:

- Can you just change a setting in a module or permissions?
- Can you accomplish your goal through CSS?
- Just replace a string?
- Can you modify output in Views?

Session 4: Advanced theming introduction

You can get a long way theming in Drupal with very little PHP coding, as we have seen using the templating system. However, there are sometimes you want to modify the way output is rendered. Later, we're going to see some easier ways to do this, but for now, we'll do it the hard way, so you can learn more about Drupal's underpinnings.

Unit 4.1 - Coding standards in Drupal

Drupal coding standards are particularly important when you decide to contribute your theme. However, sticking to consistent coding standards in your theme will help when you collaborate with your colleagues.

- Drupal Coding standards <http://drupal.org/coding-standards>
- Drupal CSS coding standards <http://drupal.org/node/302199>
- Drupal comment standards <http://drupal.org/node/1354>

Notable formatting standards for themers

- Use an indent of 2 spaces, with no tabs.
- Lines should have no trailing whitespace at the end.
- Files should be formatted with \n as the line ending (Unix line endings), not \r\n (Windows line endings).
- In general, all lines of code should not be longer than 80 chars.
- In CSS, multiple properties should be listed in alphabetical order.

Unit 4.2 - Basic structures in PHP

Drupal is written in PHP, which was originally a templating language. This improved web development greatly. Developers could save time reusing the same "header" code throughout their site. Templates allowed for them to write once and reuse. In this session you'll get more familiar with the terminology.

- **Variables:** Allowed developers to pass information. For example the "current page title" to a header template or "number of visitors".
- **Control structures:** Allowed developers to use logic. For example as "if this is the front page, show the big logo".
- **Functions:** Allowed developers to pass data or information, and manipulate it or evaluate it and send back a result or outcome. For example "take the current count of visitors and add one more and return the new count value".
- **Arrays:** This allowed for structured data to be passed. For example, the contents of an entire page could be output as an array: "title, body, tags" instead of having to pass each separately.

If you're familiar with other programming languages PHP will look similar to any languages following the C style syntax. Using data and control structures like these, PHP made it easier to serve dynamic pages and also have script-level programming to manipulate data. With database interaction, developers could make simple web applications. We'll look in more detail now to how these are implemented in the theme layer in Drupal. We will make references to the relevant PHP documentation <http://www.php.net/manual/en/index.php>

Unit 4.3 - PHP Basics, Functions

This may seem backward, but to understand the templating system in Drupal, you will first need to use functions. Your instructor will do a quick introduction to functions in PHP. Read more:

- PHP documentation, Functions <http://www.php.net/manual/en/language.functions.php>
- Drupal documentation on theme functions <http://api.drupal.org/api/drupal/modules--system--theme.api.php/group/themeable/7>

```
function add (a, b) {  
    c = a + b;  
    return c;  
}
```

Unit 4.4 - Using functions in Drupal

You use functions all the time in Drupal to pass values to functions which are like sub-programs within Drupal.

Exercise: Add template.php to your theme

1. Create a text file named "template.php" file in your theme's main folder.
2. Type in the opening tag which indicates this is the start of PHP code
`<?php`
... notice you **do not close** that tag!
3. Add a new function with opening and closing curly braces.
4. Save the file

Exercise: Add a theme function for breadcrumbs

A. About the breadcrumb

The breadcrumb contains a link back to parent links. For example, go to edit an item of content. You can see the breadcrumb appearing under the menu.

The screenshot shows a Drupal page titled "Edit Basic page About us". At the top left, there is a breadcrumb trail: "Home > About us". Below the title, there are two buttons: "View" (highlighted with a blue border) and "Edit".

There are several modules to help you configure breadcrumb trails.

- Hansel <http://drupal.org/project/hansel> to manipulate breadcrumb trails.
- Context <http://drupal.org/project/context> we'll see how to set breadcrumbs using context.

The breadcrumb is formed through a theme function which defines how it is output. You can find the relevant line by opening up your Drupal core's file includes/theme.inc

http://api.drupal.org/api/drupal/includes--theme.inc/function/theme_breadcrumb/7

B. Override the default breadcrumb code

1. Copy the theme function from the core folder into our template.php file. Open includes/theme.inc, and search for themeBreadcrumb. Copy this function, ending with the curly

```

bracket }

function theme_breadcrumb($variables) {
  $breadcrumb = $variables['breadcrumb'];

  if (!empty($breadcrumb)) {
    // Provide a navigational heading to give context for breadcrumb links to
    // screen-reader users. Make the heading invisible with .element-invisible.
    $output = '<h2 class="element-invisible">' . t('You are here') . '</h2>';

    $output .= '<div class="breadcrumb">' . implode(' » ', $breadcrumb) . '</div>';
    return $output;
  }
}

/**

```



- Paste this into your theme's template.php file, and rename the function name to match your theme's name*. As shown here:

```

<?php

function mytheme_breadcrumb($variables) {
  $breadcrumb = $variables['breadcrumb'];

  if (!empty($breadcrumb)) {
    // Provide a navigational heading to give context for breadcrumb links to
    // screen-reader users. Make the heading invisible with .element-invisible.
    $output = '<h2 class="element-invisible">' . t('You are here') . '</h2>';

    $output .= '<div class="breadcrumb">' . implode(' » ', $breadcrumb) . '</div>';
    return $output;
  }
}

```

* When you override theme functions, you always replace theme_ with **your theme name**.

- Make a visible change, such as changing the » symbol to /

```

$output .= '<div class="breadcrumb">' . implode(' » ',
$breadcrumb) . '</div>';

```

- to

```

$output .= '<div class="breadcrumb">' . implode(' / ',
$breadcrumb) . '</div>';

```

- Save the file.

- Clear your cache.** Because you're adding a new file to your theme, you need to not only register this new template file, but also this function.

- Refresh the page, and you see the new breadcrumb format. In this case we're showing the breadcrumb when you edit a page titled "About us".



Home / About us

Unit 4.5 - PHP Basics, Variables

In order to pass information to and from functions you will need to use variables. You instructor will do a quick introduction to variables in PHP. Read more:

- PHP documentation <http://www.php.net/manual/en/language.variables.php>
- Search Drupal.org documentation for specific variables <http://api.drupal.org/api/drupal/7>

```
a = 3;  
b = 2;  
c = a + b;
```

Unit 4.6 - Variables in Drupal

When you set or reset your username in Drupal, you're changing the value of the variable for your username. At this point, you may already have a concept of what variables are. In this short activity, you'll see how to set and retrieve variables in Drupal.

While editing the template files you may have noticed a number of variables being printed out. These return full HTML output.

```
<?php if ($breadcrumb): ?>
  <div id="breadcrumb"><?php print $breadcrumb; ?></div>
<?php endif; ?>

<?php print $messages; ?>
```

For example, the message might print out as

```
<div class="messages warning">
  <h2 class="element-invisible">Warning message</h2>Don't panic!
</div>
```

To create or modify these kinds of variables, you can employ preprocess functions.

Exercise – Add a variable

A. Add a preprocess page function

Create your theme_preprocess_page function. Make sure the function name is the same name as your theme.

```
function mytheme_preprocess_page(&$variables) {
}
```

B. Set the value for the variable

Create a new variable for 'credits'.

```
function mytheme_preprocess_page(&$variables) {
  $variables['credits'] = t("Built by me!");
}
```

C. Add variable to page.tpl.php

Place your new variable in an appropriate part of the page template with

```
<?php print $credits; ?>
```

Unit 4.7 - PHP Basics, Arrays

As you see, you often pass more than one string of text in a variable. Arrays help to structure that data. Your instructor will do a quick introduction to Arrays in PHP. Read more:

- PHP documentation, Arrays <http://php.net/manual/en/language.types.array.php>
- Render arrays in Drupal <http://drupal.org/node/930760>
- Screencasts about Render arrays linked from here: <http://drupal.org/node/1043866>

Understanding arrays

- Each row is an array, each seat a container.
- Look at the people in your row.
- The index starts with 0. What's your position?



<http://www.flickr.com/photos/jronaldlee/4524199846/>

Unit 4.8 - Render arrays - How Drupal creates the HTML string

In this unit we'll see how Drupal's theme layer creates the markup for your page.

In Drupal 6, some items were simply difficult to move from different "vertical stacks", such as from the node stack to the block stack. Since then a new functions hide() and render() allows you to print out selected items from nodes or profiles.

More about Render Arrays in Drupal 7 <http://drupal.org/node/930760>

Using hide() and render()

A. Prepare your site

- The article content type contains an image field. Add an article with an image to test with.
- Enable the **Devel** module <http://drupal.org/project/devel>

B. Discover the elements in the content array

1. Open the template.php file for your theme (created in the previous session). Add the following preprocess function into the template.php file. Make sure the function name "mytheme" matches the name of your theme.

```
function mytheme_preprocess_node (&$variables) {  
    kpr($variables);  
}
```



Alternative methods of getting the array structure

To also get the structure , you can try :

```
print_r($variables)
```

Examples:

```
print_r($node);  
print_r($content);
```

You can also add kpr() directly to template files with this:

```
<?php kpr(get_defined_vars()); ?>
```

2. Save the file and clear your cache. Like the .info file, the template.php file is cached. When you add new functions you need to clear your cache.
3. Visit the article to which you have uploaded an image. This will generate a click-able display of your node content.

4. Scroll down to locate the "content" array and click to open it.

node_uri (String, 24 characters) /content/use-a-tortoise-or-an-erat
page (Boolean) TRUE
content (Array, 4 elements)
field_image (Array, 16 elements)
comments (Array, 4 elements)
links (Array, 5 elements)
body (Array, 16 elements)
display_submitted (Boolean) TRUE
submitted (String, 252 characters) Submitted by
user_picture (String, 0 characters)

5. Now you can see what is being output.

C. Control the display with hide() and render()

Comments, links and images can be excluded from the output initially, then rendered afterwards. There you can control how they are displayed.

1. Optional in case you don't already have this file. Open the default node.tpl.php. Save a copy as node--article.tpl.php
2. Locate the start of div class="content" and replace the remainder of the node template with this code.

```
<div class="content"<?php print $content_attributes; ?>>
<?php
    // We hide the comments and links now so that we can
    render them later.

    hide($content['comments']);
    hide($content['links']);
    hide($content['field_image']);
    print render($content);
?
</div>
<div style="float:right; margin:0 0 0 20px">
<h3>This is an image
    <?php print render($content['field_image']); ?>
</h3>
</div>
    <?php print render($content['links']); ?>
    <?php print render($content['comments']); ?>
</div>
```

3. Save your template file and view your specific article. Notice how this changed the display of the image.
4. What are some ways you could use this to control the display of content on your site?

Session Summary

In this session we saw the theming layer in more detail. If you're new to it, you got an introduction to some basic PHP coding. Much of what you can do at this level of coding in Drupal can also be done through configuration, saving you time and making your sites easier to upgrade.

Sometimes however, you need to go further than even the examples above, and you may need a custom module.

Do you need a module?

- *Do you need to access the database directly? SQL should never be in a theme.*
- *Are you rebuilding data?*
- *Should your changes be available even if you were to disable your theme?*

If the answer to any of these questions is yes, then your changes belong in a module.

Summarized from **Definitive Guide to Drupal**. Chapter 16: Advanced Theming > Sustainability and Best Practices - Pg. 354

Session 5: Controlling layout through configuration

From the first point of site planning, take into account the design. Always use a 'configure first' approach to theming. We'll talk about how you can do as much through configuration and make your theme and site easier to upgrade.

- Build a *content type*, to model data and identify the components, but also the display of the content.
- Build a *view* to control not only if something is displayed as a block, page or attachment, but also what format the content takes: HTML list, table, or unformatted.
- Control *block visibility* through configuration, you control what appears on your site and where.

If you're developing your own modules or working with developers, make sure to take into account the customization of appearance. We'll look at some contributed modules to see how to theme with and for them.

Unit 5.1 - Site building tips

Modules output their own markup, you can only find this after configuration. You may find with contributed modules, there are often inconsistencies in the markup. It's assumed you already have essential site building skills, but it's a good time to make a review. Your instructor will do a demo of site building techniques to improve the output markup including:

- **Manage display:** Setting field display and view modes for a content type.
- **HTML settings in Views:** Edit the output of Views.
- **Media > Image styles:** Edit image styles.

Can you think of situations you have had before where you were unable to get the result you wanted through configuration?

Unit 5.2 - Contributed modules for theming

When considering the look and feel of your site, you may be mainly thinking on the macro level of layout and positioning of content or interaction. There are many helper modules which will make it easier to get more fine-grained control of the look and feel. Always check first what Drupal can do out of the box. For example:

- Change date format in Configuration > Regional and language > Date and time > Formats
- Change automatic image resizing in Configuration > Media

Modules to make theming easier

There are modules such as Devel and Styleguide to help improve the developer experience for themers. These modules can speed up your time to get just the interactivity you want.

Module	Info
Beauty Tips	http://drupal.org/project/beautytips Provides balloon-help style tooltips using jQuery.
@font-your-face	http://drupal.org/project/fontyourface An administrative interface for browsing and applying web fonts (using CSS @font-face from various sources).
jCarousel	http://drupal.org/project/jcarousel Turn any list of content or images into a carousel jCarousel with jQuery.
Lightbox2	http://drupal.org/project/lightbox2 Displays images, forms, or content in a popup overlay.
Menu block	http://drupal.org/project/menu_block A great module which can put a main menu in one region, such as across the top of a site, and the child items in a sidebar menu.
Nice Menus	http://drupal.org/project/nice_menus Enable drop-down, expandable menus. Uses jQuery by default, but has CSS fallbacks for older browsers.
Quicktabs	http://drupal.org/project/quicktabs The Quick Tabs module allows you to create blocks of tabbed content
String overrides	http://drupal.org/project/stringoverrides Change "Create new account" to something more chirpy and friendly?
Superfish Menus	http://drupal.org/project/superfish Create fly out menus with the superfish jQuery plugin
Views Accordion	http://drupal.org/project/views_accordion Display content in Views as a jQuery accordion, where the first field expands when clicked to reveal the other fields.

Module	Info
Views Slideshow	http://drupal.org/project/views_slideshow Display content in Views as a jQuery slideshow, with many customization options.

Optional Exercise – Views slideshow

If you've never used a module to extend Views before, this is a great example. In this exercise we'll add a slideshow to our sidebar with **Views Slideshow**

A. Prepare your site

- You should have some images in articles from earlier. If not, add at least 3 articles with images.
- Install and enable Views Slideshow module.
http://drupal.org/project/views_slideshow
- Install and enable Libraries module.
<http://drupal.org/project/libraries>
- Install the **jQuery cycle plugin**. Create a directory in sites/all/libraries called jquery.cycle, and then copy jquery.cycle.all.min.js into it. Download the plugin at <http://malsup.com/jquery/cycle>

B. Add a Views slideshow

1. Go to Structure > Views and + Add new view. Configure your view with the following settings:
 - Show content of type Article tagged with [leave empty] sorted by Newest first
 - De-select Create a page
 - Select Create a block.
 - Title: leave default
 - Display format: Slideshow of fields

Welcome

Ut aliquam condimentum lacus; id vehicula nisi tempus eu. Aliquam erat volutpat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

Sample slideshow



- Items per page: 3

Home » Administration » Structure » Views

Add new view [Create a page](#)

View name *
Sample slideshow Machine name: sample_slideshow [Edit]

Description

Show Content of type Article tagged with sorted by Newest first

Create a page

Create a block

Block title Sample slideshow

Display format Slideshow of fields
 teasers
 full posts
 titles
 titles (linked)
 fields

Items per page 3

Save & exit Continue & edit Cancel

2. Click Continue & Edit.

3. In Views, first column click to configure Slideshow | **Settings**. This opens a dialogue window. Explore the options for *transitions* and choose one.

Displays

Block + Add edit view name/description clone block

Block details

Display name: Block

TITLE
Title: Sample slideshow

FORMAT
Format: Slideshow | **Settings** Change settings for this format

Show: Fields | Settings

FIELDS add
Content: Title

FILTER CRITERIA add
Content: Published (Yes)

BLOCK SETTINGS
Block name: None
Access: Permission | View published content

HEADER add
FOOTER add

PAGER
Use pager: Full | Paged, 3 items
More link: No

Advanced

4. Under Fields add the *image field* from the article.

- On Configure field choose Image style: *medium*. This will be small for our sidebar, you can add a custom image style if you want.
- Click to expand Style settings. Notice you can add classes to the image and customize the HTML. No changes needed at this time.
- Click Apply (all displays).

5. Click the title field to open the options window, and remove the title field.

6. Next we'll filter out only results that have images so we don't get an empty slide.

- Click to Add filter criteria.
- Select Content: Image (field_image:fid)
- Click Add filter. Choose Operator: *Is not empty (NOT NULL)*.

The screenshot shows the 'Configure filter criterion' dialog for 'Content: Image (field_image:fid)'. At the top, there's a dropdown menu labeled 'For' set to 'All displays'. Below it, a note says 'Appears in: node:article.' There's a checkbox for 'Expose this filter to visitors, to allow them to change it'. Under 'Operator', a dropdown menu is set to 'Is not empty (NOT NULL)'. A 'MORE' link is available for additional operators. At the bottom, there are three buttons: 'Apply (all displays)', 'Cancel', and 'Remove'.

7. Click **Apply (all displays)**. Finally, **Save** the view.
8. Go to Structure > Blocks and add this new block to your sidebar.

Challenge Exercise – Try another jQuery module

Choose from the list above or do a search on Drupal.org for a module (compatible with Drupal 7) which has jQuery support. See what kind of customization you can do through configuration or through the theme layer.

Modules for layout and positioning

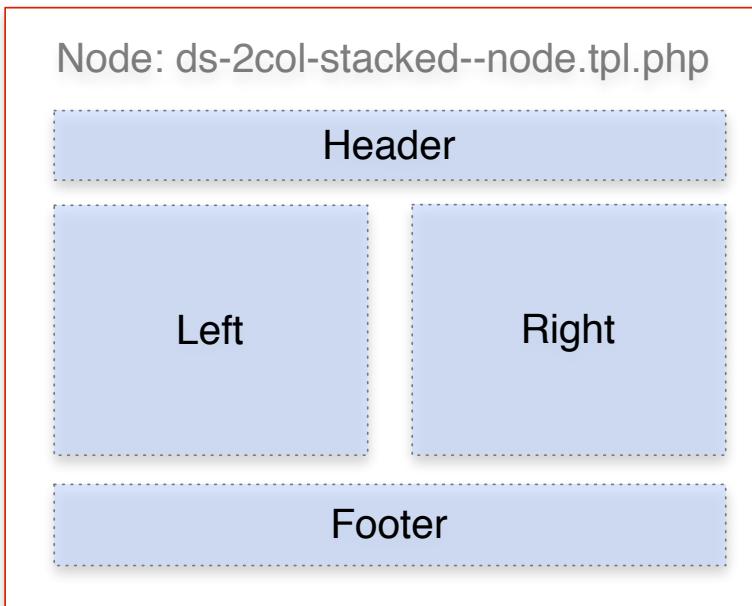
At some point you may want to control something a little more than the block system will allow. There are several solutions which suit different needs. We'll look at these in more detail in the next unit.

- **Display suite** <http://drupal.org/project/ds> Get control over node display. Add regions within a node
- **Panels** <http://drupal.org/project/panels> Use a drag and drop interface to control over display of block-type content based on specific conditions and file paths.
- **Page Manager** <http://drupal.org/project/ctools> This is part of the Chaos Tools suite to manage pages.
- **Context** <http://drupal.org/project/context> Manage contextual conditions and reactions for different portions of your site. You can think of each context as representing a "section" of your site.

Unit 5.3 - Display suite for custom teaser

About Display suite

Display suite <http://drupal.org/project/ds> allows you to divide your node content into "regions", much like you do with block placement. With default options available in "manage display" for a content type, you can only arrange some parts of a node. Fields, yes, but the Read more link? No. Display suite allows you to control the placement of all elements output in the node template.



Exercise – Custom article teasers

A. Prepare the site

1. Enable the Display suite module, Views and the Views UI, if you don't have them currently enabled.
2. Go to Structure > Views.
3. Locate the disabled Archive view, and click "enable". You should now see Archive as an active view. Click the /archive path. This takes you to a list of months. When you click on the month you can see the default teaser view is currently being used. We're going to make a custom View mode to use instead.

The screenshot shows the Drupal Views UI. At the top, there are links for 'Home', 'Administration', and 'Structure'. Below that is a navigation bar with tabs for 'Views' (which is selected), 'LIST', and 'SETTINGS'. Underneath, there are buttons for '+ Add new view' and '+ Import'. A table lists the views, with the 'Archive' view highlighted. The table columns are: VIEW NAME, DESCRIPTION, TAG, PATH, and OPERATIONS. The 'Archive' view has the following details: DISPLAYS: Block, Page; In code; Type: Content; Description: 'Display a list of months that link to content for that month.'; TAG: default; PATH: /archive; and OPERATIONS: an 'edit' button with a dropdown arrow. The 'SETTINGS' tab is also visible.

VIEW NAME	DESCRIPTION	TAG	PATH	OPERATIONS
Archive Displays: Block, Page In code Type: Content	Display a list of months that link to content for that month.	default	/archive	edit ▾

B. Add a View mode

1. Go to Structure > Display Suite > View modes. Click + Add a view mode.
2. Configure with the following settings.
 - Label: Mini teaser
 - Entities: Select Node

Home » Administration » Structure » Display suite » View modes

View modes •

Label *
Mini teaser Machine name: mini_teaser [Edit]
The human-readable label of the view mode. This name must be unique.

Entities *
 Comment
 Node
 Taxonomy term
 User
Select the entities for which this view mode will be made available.

Save

3. Click Save.

C. Modify the article layout.

1. Go to Structure > Content types > Article > Manage display
2. Scroll down to Custom display settings.
 - Select **Mini teaser**.

Custom display settings	Use custom display settings for the following view modes
Layout for article in default	<input type="checkbox"/> Full content
Clone layout	<input checked="" type="checkbox"/> Teaser
	<input type="checkbox"/> RSS
	<input type="checkbox"/> Search index
	<input type="checkbox"/> Search result
	<input checked="" type="checkbox"/> Mini teaser
	Manage view modes

- Click Save.

3. Click the Mini teaser view mode tab.

4. Currently, you can only arrange fields such as the image, body and tags.

FIELD	LABEL	FORMAT	
⊕ Image	<Hidden>	Image	Image style: medium Linked to content
⊕ Body	<Hidden>	Summary or trimmed	Trim length: 600
⊕ Tags	Above	Link	
Hidden			
<i>No field is hidden.</i>			

5. Scroll down, you'll notice the View modes options are changed. Under the **Layout for article in teaser** tab, select Two column stacked.

6. Click Save.

D. Arrange fields

1. You now have regions available, Header, left, right, footer. Arrange the fields under these regions:

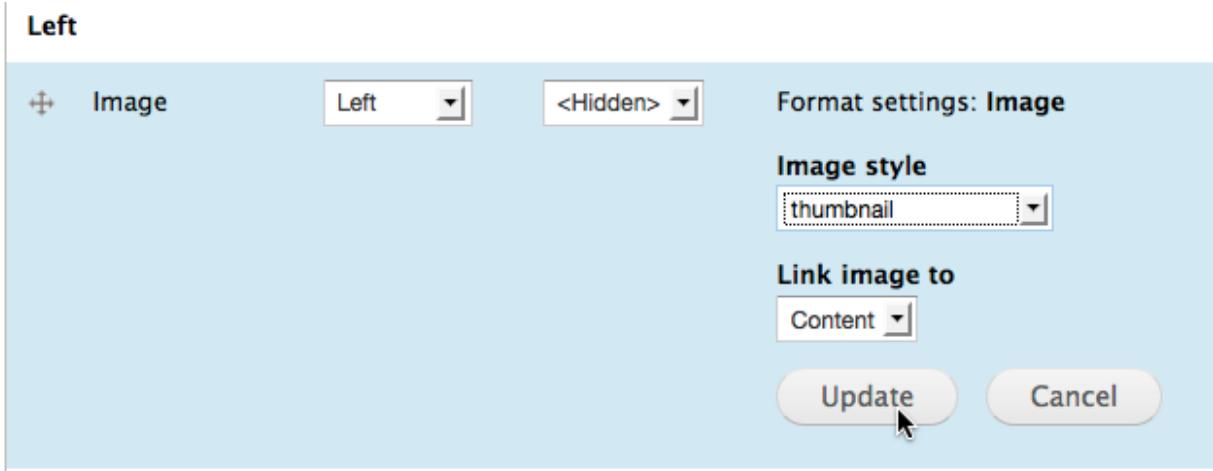
- **Header:** Title
- **Left:** Image.
- **Right:** Body.

FIELD	REGION	LABEL	FORMAT	
Header				
⊕ Title	Header	<Hidden>	Default	Wrapper: h2
Left				
⊕ Image	Left	<Hidden>	Image	Image style: large
Right				
⊕ Body	Right	<Hidden>	Summary or trimmed	Trim length: 600

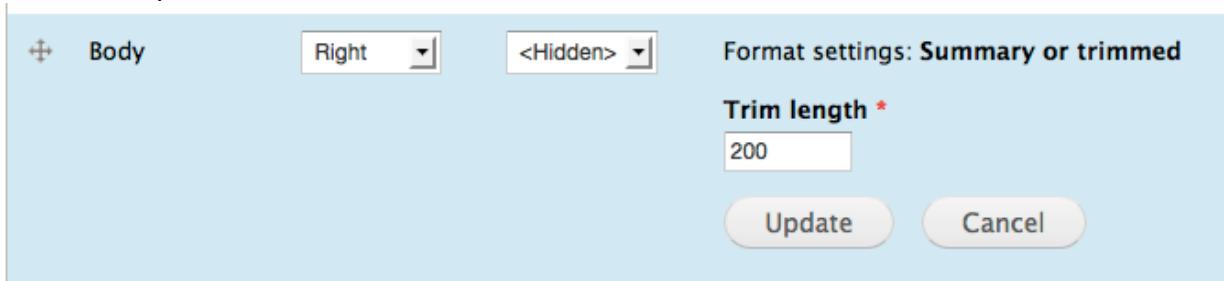
- Keep in mind, this is only the teaser, so we don't want comments, etc to display.

E. Configure fields

- Under Image format, click the gear symbol. Change Image style to thumbnail. Click Update.



- Under Body Format, Select Summary or trimmed. Click the gear and change Trim length to 200. Click update.



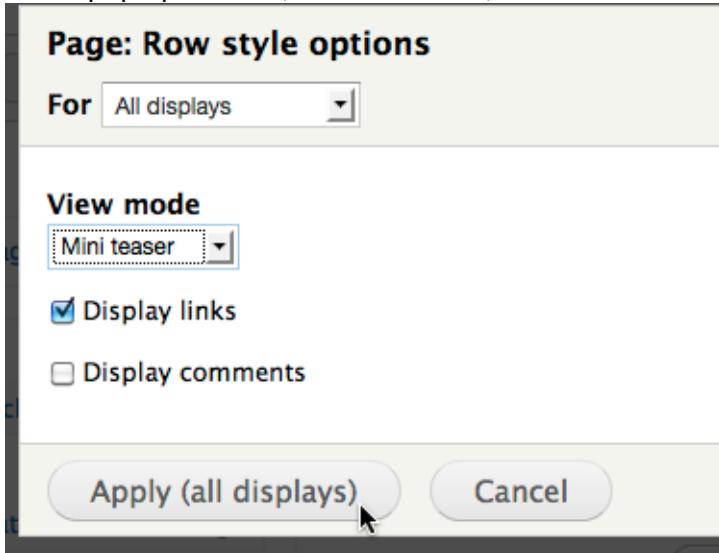
- Click Save.

F. Edit the Archive view

- Go to edit the archive view. In the first column, under Format, click Teaser to edit the settings.

The screenshot shows the 'Displays' configuration for the 'Page' view. At the top left are tabs: 'Page' (selected), 'Block', and '+ Add'. To the right is a 'edit view name/description' dropdown. Below this is a section titled 'Page details' with a 'Display name: Page' field and a 'view page' button. The main area is divided into three columns: 'TITLE' (Title: Monthly archive), 'PAGE SETTINGS' (Path: /archive, Menu: No menu, Access: None), and 'HEADER' (with an 'add' button). At the bottom left, under 'FORMAT', it says 'Format: Unformatted list | Settings' and 'Show: Content | Teaser'. The 'Teaser' link is highlighted with a blue box and a cursor arrow pointing to it.

2. In the pop up window, for View mode, select Mini teaser.

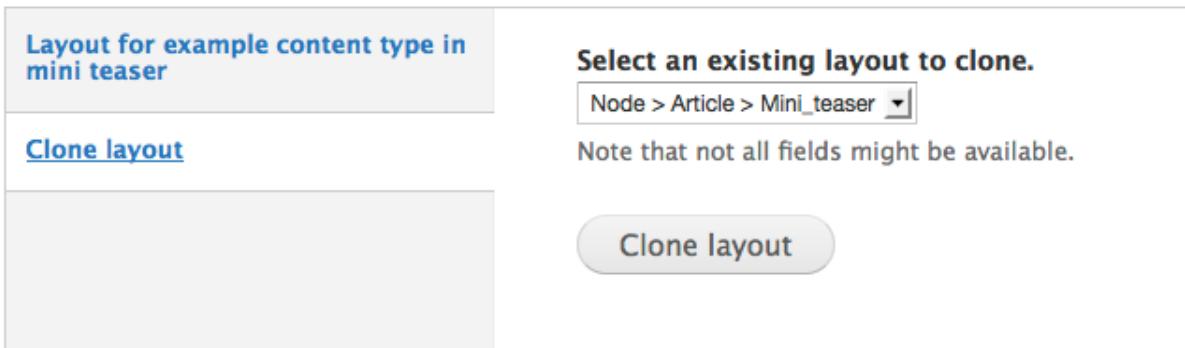


3. Save the view. And review the changes at /archive

G. Optional: Clone the layout to other content types

You may notice at /archive, if you have other content types, they fall back on using the default teaser view. If you have other content types, you can clone the layout used in the Article.

1. Enable the Mini teaser mode.
2. Go to the Mini teaser tab.
3. Scroll down to "Clone layout". Select Node > Article > Mini_teaser.



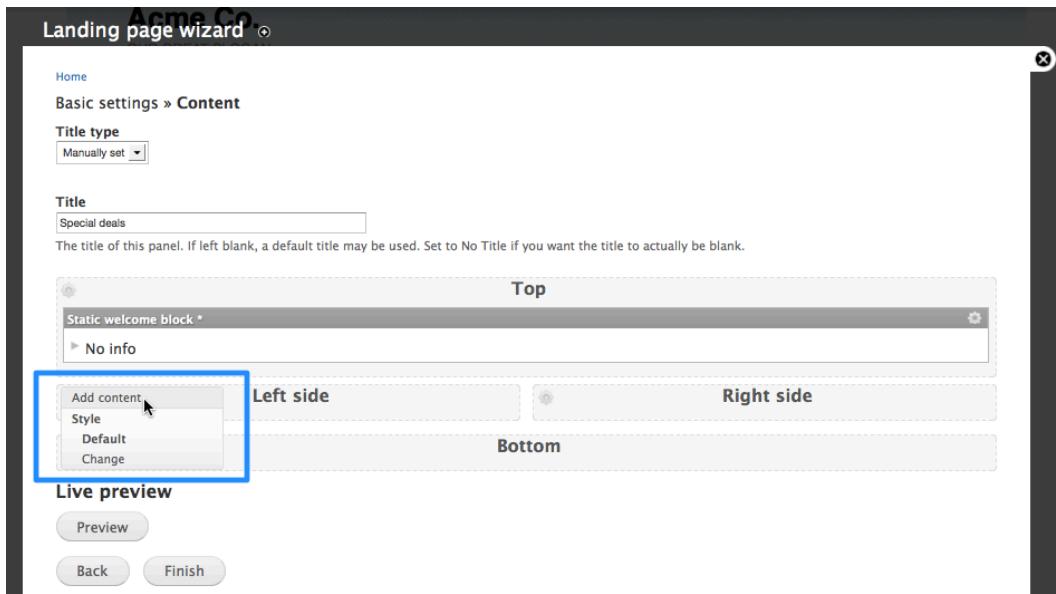
4. Click Clone layout.
5. Double check the format settings for the fields. Especially if you have different fields available.
6. Click Save. You can use Display suite as a way to get custom displays for specific custom views.

Unit 5.5 - Panels for custom layouts

Panels is a powerful and complex drag and drop layout editor. <http://drupal.org/project/panels>

About Panels

There are several layouts included with Panels, however you can use the layout design tool to create your own. Panels also provides a useful drag and drop interface which can be handed over to other team members or a client. An example scenario would be a trained marketing team, creating or editing specific landing pages for initiatives or promotions.



Unit 5.4 - Context module for site sections

Home

About

Articles

In this activity, you will see the active menu item for various situations.

Exercise – Prepare the site

A. Add content and the necessary modules

- You should have a number articles in your site already.
- Go to Content. Add a "Basic page" titled "About" and add this to the main menu.
- Add and enable Context and Context UI modules. <http://drupal.org/project/context>

CONTEXT					
ENABLED	NAME	VERSION	DESCRIPTION	OPERATIONS	
<input checked="" type="checkbox"/>	Context	7.x-3.0-beta2	Provide modules with a cache that lasts for a single page request. Requires: Chaos tools (enabled) Required by: Context layouts (disabled), Context UI (enabled)	Help	
<input type="checkbox"/>	Context layouts	7.x-3.0-beta2	Allow theme layer to provide multiple region layouts and integrate with context. Requires: Context (enabled), Chaos tools (enabled)		
<input checked="" type="checkbox"/>	Context UI	7.x-3.0-beta2	Provides a simple UI for settings up a site structure using Context. Requires: Context (enabled), Chaos tools (enabled)	Help	Configure

B. Add a view.

1. Add a view titled "Articles". listing article teasers. This will make a view at the path /articles

Add new view

Home » Administration » Structure » Views

View name *

Articles

Machine name: articles [\[Edit\]](#)

Description

Show Content

of type Article

tagged with

sorted by Newest first

2. By default *Create a page* is selected. Make sure it is an "unformatted list of teasers".

3. Select **Create a menu link**. Make sure Main menu is selected. Keep other defaults.

The screenshot shows the 'Create a page' configuration form. It includes fields for Page title (Articles), Path (http://d7front.localhost:8082/articles), Display format (Unformatted list, teasers, with links, without comments), Items to display (10), Use a pager (checked), Create a menu link (checked), Menu (Main menu), Link text (Articles), and Include an RSS feed (unchecked). A cursor arrow is visible on the right side of the form.

Create a page

Page title
Articles

Path
http://d7front.localhost:8082/articles

Display format
Unformatted list of teasers with links (allow users to add comments, etc.) without comments

Items to display
10

Use a pager

Create a menu link

Menu
Main menu

Link text
Articles

Include an RSS feed

4. Scroll down and click **Save and exit**. You will see your "Articles" item in the main menu.

Exercise – Create and configure a Context

Notice when you click on the Articles top menu item, the menu item appears "active". Inspect the source code and see what is controlling that. We'll make it so when you click on an Article the Articles menu item will stay active.

1. Go to Structure > Context + Add

- Name: article_section
- Tag: Sections
- Description: To highlight the section on the main menu

- Under **Conditions** > Add conditions. Select *Node type*, and select *Article*.

The screenshot shows the 'Conditions' section with a trigger 'Trigger the activation of this context' and a button '**<Add a condition>**'. Below it is the 'Node type' section with two options: 'Article' (checked) and 'Basic page'. A note says 'Set this context when viewing a node page'. At the bottom is the 'Set on node form' section with a dropdown set to 'Yes' and a note 'Set this context on node forms'.

- Add another **Condition**. Select *Path*. And type in articles/* in the Path field.
- Under **Reactions** > Add a reaction. Select *Breadcrumb*, and select *Main menu -- Articles*.
- Add another **Reaction**. Select *Menu* and select *Articles*. This sets the active class on an item.
- Now, click **Save**.
- Check your site. Is the article menu item highlighted? Is it highlighted when it goes to the articles? What is missing? We don't have the .active class styled.
- Edit your navigation.css file. At the bottom, edit the CSS which controls display and add `#navigation li a.active` to stack the selectors. Your CSS file might look like this:

```
#navigation li a:hover, #navigation li.active-trail a, #navigation li a.active {
    background-color: #4e8fac;
    text-decoration:none;
    color:#ffffff;
    font-size:bigger;
}
```

- Save your CSS changes and refresh your site.
- As you have started to see, Context is a powerful module to tie parts of your site together. Try to make another Context and see the variety of conditions and reactions you can have. Finally, as you saw you also need to make some alterations in your theme to accommodate Context.

Session summary

In this session we saw some powerful modules for controlling layout, and in some ways, completely overriding how Drupal's core block system works. Context and Panels are robust modules, however- if you're only using them in a few situations, they might be overkill. Display suite is applicable in many situations allowing you to get more control over the display of content.

Session 6: Responsive theme systems

So far we've seen how a basic Drupal theme, even with an .info file inherits the templates of the core and contributed themes. Now we'll see how you can customize further with sub-themes and base themes.

Unit 6.1 - What are sub-themes?

You should never alter the code of any module or theme downloaded from Drupal.org. If you want to alter the way a contributed theme looks, you should make a sub-theme.

Sub-theme best practice

A sub-theme inherits the template files, CSS and settings of an identified parent theme.

- Can even be a sub-theme of another sub-theme

CSS and JS are inherited, but can be overridden

- Reusing the same filename removes the original stylesheet

Sub-themes do not inherit custom regions

- Header, Content, First and Second Sidebar, and Footer are defaults

Exercise - Explore a sub theme example

If you haven't already, download the **Cornflower** and **Jasmin** themes from <https://github.com/acquia/training>. Place them in your site/all/themes... folder.

In this example, Jasmin is the sub-theme to Cornflower.

A. Enable and compare the themes

1. Enable the **Cornflower** theme and set as default.
2. Go to an article, and look at the view/edit tabs. What changes can you see?
3. Enable the **Jasmin** theme and set as default.
4. Refresh your site. What differences do you see?

B. Compare the files

1. Compare the .info files for Cornflower and Jasmin themes.
2. Compare the CSS files available
3. What is being overridden in the Jasmin theme?

C. Conclusions

What do you think were the rules of inheritance between the themes?

Consider: **System's CSS, Base theme CSS and Child theme CSS.**

Unit 6.2 - What are base themes?

Developers started to leverage the sub-theme functionality. They would create generic, abstracted base themes which would take care of the most common needs they had. They could employ a sub-theme to make customizations. Good thing is, these developers share their base themes!

Popular base themes

There are a number of base themes which make good starting points. Creating sub-themes to these systems can make it much easier to create maintainable themes. As with all Drupal projects, always read the README.txt file that comes with these projects.

Making a truly responsive theme for your site takes time and testing. Base themes get you up and going quickly, and save you some time and headaches.

- Adaptive <http://drupal.org/project/adaptivetheme> A responsive HTML 5 theme
- Omega <http://drupal.org/project/omega> A responsive mobile-first HTML 5/960 theme
- Fusion <http://drupal.org/project/fusion> A configurable base theme
- Genesis <http://drupal.org/project/genesis> A HTML 5 theme with rich CSS support
- Zen <http://drupal.org/project/zen> A theme full of classes for easy CSS customization

HTML 5 support

HTML 5 support for Drupal is currently being developed. Because both core and contributed modules add markup, reconfiguring Drupal for HTML 5 requires additional work. Find out more:

- HTML 5 group <http://groups.drupal.org/html5>
- HTML 5 Base theme http://drupal.org/project/html5_base
- HTML 5 Tools http://drupal.org/project/html5_tools

Unit 6.3 - A quick demo of the Adaptive system

In this unit we'll explore the Adaptive theme system, and install a sub-theme, Corolla. Corolla is fairly easy to adapt through configuration, and you can use the tools to change the colors, and add rounded corners, change fonts and more.

Exercise: Install a sub-theme

A. Install the base theme

Install the Adaptive system <http://drupal.org/project/adaptivetheme>

To use a sub-theme you do not need to enable the related base theme.

B. Install and enable Corolla

- Corolla <http://drupal.org/project/corolla> See a demo: <http://adaptivethemes.com/demos/contrib/?theme=corolla>
- Alternatively you can also try Sky <http://drupal.org/project/sky> See a demo: adaptivethemes.com/demos/contrib/?theme=sky

Explore theme settings

- Go to Appearance > Settings and click the settings for Corolla.
- Click Standard layout. Here you can change sidebar positions.

The screenshot shows the 'Standard Layout' settings for the Corolla theme. On the left, a sidebar lists various theme settings: Tablet Layout, Smartphone Layout, Fonts, Font Size, Heading Styles, Rounded corners, Box Shadows and Textures, and Image Settings. The main area is titled 'Standard Layout' and contains the following sections:

- Choose sidebar positions:** Three options are shown: Left (blue bar on left), Right (blue bar on right), and Both (blue bars on both sides). The 'Right' option is selected.
- Set the width of each sidebar:** A unit dropdown is set to '%'. The 'First sidebar' is set to 20% and the 'Second sidebar' is set to 25%.

- Click Smartphone layout. Here you can opt to stack sidebar, or put them side-by-side. You can also directly edit the media query.

Smartphone Layout

Smartphone devices such as iPhone, Android and Wind for landscape orientation only – portrait orientation alv

Landscape smartphone

Choose sidebar positions

Set the page width

Unit % Page width * 100

Set a max width

Media query for this layout * @media

```
only screen and (min-width:321px) and (max-width:480px)
```

Do not include @media its included automatically

D. Discuss

Base theme systems offer a trade off. They take care of much of the pain of browser testing and offer a rock-solid stable foundation to build off of. Configurable themes like Corolla could allow non coders the ability to make changes to the site.

- What benefits would there be to using a base theme and sub-theme like this?
- What limitations do you find?

E. Find out more

- You can build your own Adaptive Sub themes. Refer to the **adaptivetheme_subtheme** directory in the Adaptive theme folder.
- You can also sub-theme Corolla as well. You can use Footheme to get started making a sub-theme for Sky or Corolla. See <http://drupal.org/project/footheme>

Next, we'll see how to develop our own sub-theme and review another popular responsive base system.

Unit 6.4 - Omega: A responsive theme system

Omega implements a **mobile-first** approach in the system. Omega takes care of handling the presentation of content for different displays, for example by employing jQuery and media queries to deliver the right CSS for the client's browser.

A responsive site with Omega

Check out the demo of Omega and how the site responds to resizing <http://himer.us/omega960>

When you resize to the smallest width, regions which are normally placed in a grid layout are stacked for a mobile friendly layout.

Responsive themes require preparation and planning

Some base themes have mobile or responsive support. However these will not work "out of the box". Keep in mind, if you would like a site with automatically resizing images you will need to do some preparation. Keep an eye on the development of responsive images http://drupal.org/project/responsive_images

The Omega Zones

You're quite familiar now with the idea of regions, and how these regions are controlled in the theme system. You know that region.tpl.php can be used as a base for changing the markup of regions.

Omega has "zones" within which to group regions. This helps to allow for a configurable grid system on the site.

Omega documentation

- Omega handbook: <http://drupal.org/node/819164>
- If you're new to grid systems in general, check our the 960 Grid System Basics: <http://omega.developmentgeeks.com/documentation/960grid>

Exercise – Add a sub-theme and learning about Omega

Make sure you're starting with a fresh Drupal site.

A. Modules required

- Install **Omega Tools** into your modules folder http://drupal.org/project/omega_tools
This module offers a wizard for quickly building an Omega sub-theme.

B. Set up the Omega base theme

1. Install Omega into your theme contrib folder so you have site/all//contrib/omega
2. Go to Appearance and click + Create new Omega subtheme.
3. Complete the Theme Information form with the following settings:
 - Name: myomega
 - Keep the remaining defaults
4. Click **Save and continue**.

5. Fill in a description for your theme: "This is my first Omega-based theme."
6. Click **Finish**.
This placed the new Omega theme in your theme folder. Not in "custom".
7. Locate "myomega" in your theme list. Click **Enable and set default**. You don't need to enable the Omega base theme for this to work.
8. Go to Theme settings for Omega. Scroll down to Toggle display.
 - Deselect logo.
 - Deselect secondary menu
 - Click **Save configuration**.
9. You can now disable the **Omega Tools module**.
10. View your site! **Each theme has unique regions**. You should be starting with a fresh site. However, if you're applying this theme to your old site your block will lose their placement. Your new theme doesn't have the same region names. You may need to reset them.

C. Review Omega's debugging mode

Omega comes turned on in development "debugging mode" with debugging tools and a grid overlay. All regions, whether they have blocks in them or not, appear outlined on the page.

- You can turn the grid off while viewing.
- You can turn off the block preview viewing.

If you're familiar with the 960.gs Grid System this will be welcoming and familiar to you. If you're not, you can find out more here: <http://960.gs/> and <http://www.slideshare.net/fourkitchens/accelerated-grid-theming-using-ninesixty-drupalcamp-la-2011>



When you are finished working on your theme, you will be able to turn off this debugging mode.

[Global settings](#) [Bartik](#) [My theme](#) [myomega](#) [Seven](#)

These options control the display settings for the *myomega* theme. When your site is displayed using this theme, these settings will be used.

Layout configuration

Grid settings	<input checked="" type="checkbox"/> Enable the debugging (placeholder) blocks for the selected roles. <input checked="" type="checkbox"/> Show the debugging blocks by default. This will show the debugging blocks on page load. <input checked="" type="checkbox"/> Enable the grid overlay for the selected roles. <input checked="" type="checkbox"/> Show the grid overlay by default. This will show the grid overlay on page load. Roles that may use the grid overlay and debugging blocks. <input checked="" type="checkbox"/> anonymous user <input checked="" type="checkbox"/> authenticated user <input type="checkbox"/> administrator
Zone and region configuration	
Debugging	
Toggle libraries	
Toggle styles	
Toggle advanced elements	

D. Review Grid settings Omega

1. Go to Appearance > Settings > myomega. Click Grid settings tab.
2. You'll notice you can edit the media queries for specific layouts
 - Narrow layout
 - Normal layout
 - Wide layout
3. Look at the text field for the media query setting for Narrow Layout:
 - `all and (min-width: 740px) and (min-device-width: 740px), (max-device-width: 800px) and (min-width: 740px) and (orientation:landscape)`
 - This means any device which has a screen smaller than 740px wide, will use this layout. That means any mobile device or iPad in portrait mode, for example.
4. No changes are necessary at this time.

Exercise: Prepare your site

To make the site work a little better, let's prepare the site. We've renamed this site "My Omega Site", but you can call it what you like.

A. Prepare some content and links

1. Add three basic pages and link them to the main menu.
 - About us
 - Our services
 - Contact us (enable the contact module and link to it if you want)
2. Use Devel Generate and generate 50 articles dating back 1 year.

B. Place Archive block.

1. Go to Views and enable the default Archive View. This will give us a quick block to enable.

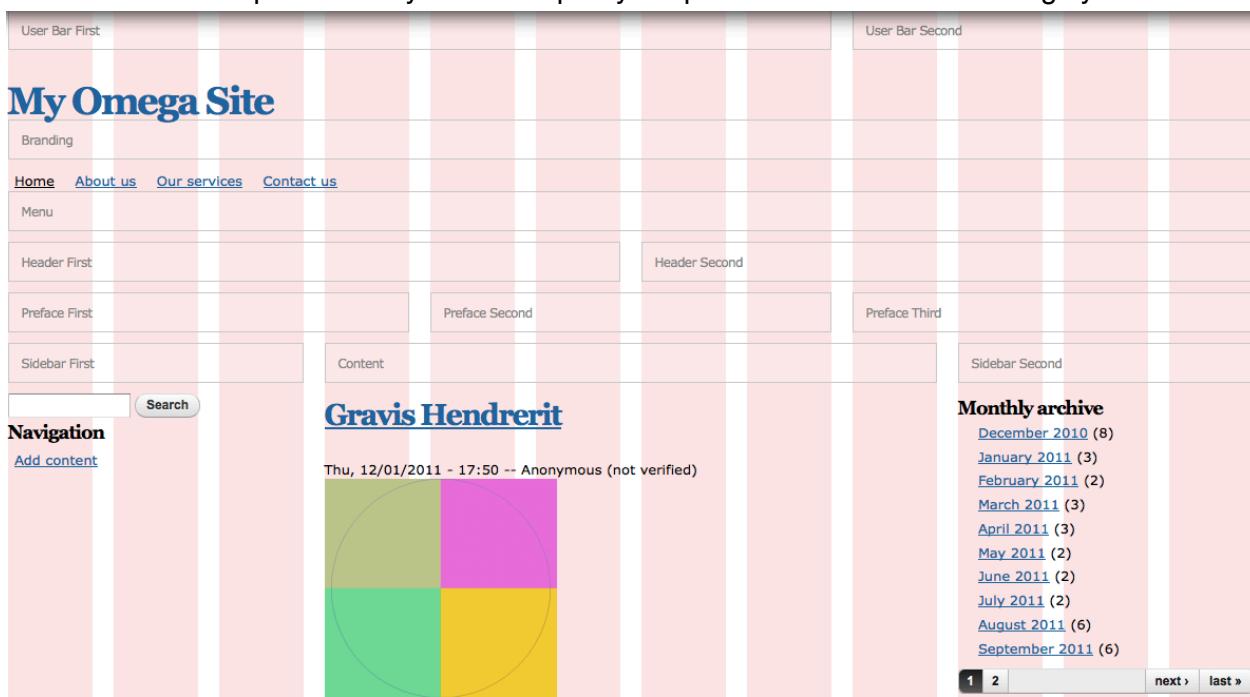
The screenshot shows the 'Views' administration page. At the top, there are links to 'Home', 'Administration', and 'Structure'. Below that, there are buttons for 'Views' (with a gear icon), 'LIST', and 'SETTINGS'. Underneath, there are buttons for '+ Add new view' and '+ Import'. A table lists two views:

VIEW NAME	DESCRIPTION	TAG	PATH	OPERATIONS
Archive Displays: Block, Page In code Type: Content	Display a list of months that link to content for that month.	default	/archive	edit enable
Backlinks Displays: Block, Page	Displays a list of nodes that link to the node, using the search	default	/node/%/backlinks	enable

2. Go to Structure > Blocks and place the new Views: Archive block in Sidebar second.

C. You are here

With all of this sample content you have a pretty simple site that should look roughly like this:



Oh, if all websites were so easy! Now we have something to work with.

Exercise: Customize the grid

In our design, we want the menu to appear to the right of the Site title. We're going to shorten the width of the menu, and place it in the Branding zone. The 960 grid system makes this super easy.



A. Adjust width of Branding

Make the width of the Branding region shorter, so the menu will be able to flow into it.

1. Go to Omega's theme settings page. Click on Zone and region configuration tab.
2. Under Header section. Click to expand > Branding zone > Regions > Branding.
3. **Width:** Select 6 columns.

Layout configuration

HEADER SECTION

- USER ZONE
- ▼ BRANDING ZONE
- CONFIGURATION
- ▼ REGIONS
- ▼ BRANDING

Force this region to be rendered
Enabling this will always render this region, even if it is empty.

Zone	Prefix	Width	Suffix	Weight
Branding	0 Columns	6 Columns	0 Columns	1
Additional region classes				

Width dropdown menu options:
1 Columns
2 Columns
3 Columns
4 Columns
5 Columns
6 Columns
7 Columns

4. Scroll down to Menu Zone and continue...

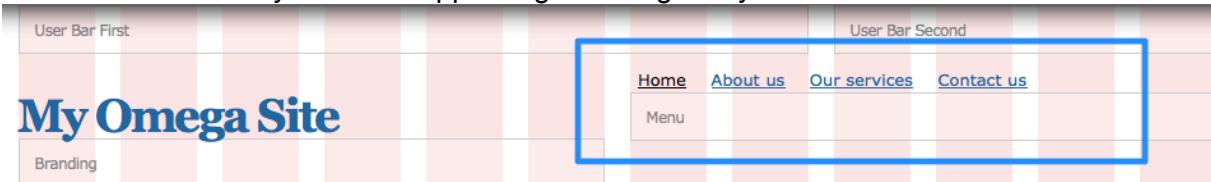
B. Reposition the menu

5. Click to expand Menu Zone > Regions > Menu. Configure with these settings:
 - Zone: Branding (this will move the menu to the branding zone after you save).
 - Width: 6 columns.

- Weight: 6. (this will ensure the menu goes to the right or after the site title).

The screenshot shows the Omega theme configuration interface. Under the 'REGIONS' section, the 'MENU' region is selected. The 'Zone' dropdown is set to 'Branding'. The 'Width' field is set to '6 Columns'. The 'Weight' field is set to '6'. The 'Force this region to be rendered' checkbox is checked.

6. Scroll down and click **Save**.
7. You should now see your menu appearing to the right of your site title!



8. Return to Omega's theme settings, and disable the debugging tools for now.
 - Under Debugging, deselect all the options.
 - Scroll down and click **Save configuration**.

Layout configuration

Grid settings	<input type="checkbox"/> Enable the debugging (placeholder) blocks for the selected roles.
Zone and region configuration	<input type="checkbox"/> Enable the grid overlay for the selected roles.
Debugging	Roles that may use the grid overlay and debugging blocks.
Toggle libraries	<input type="checkbox"/> anonymous user
Toggle styles	<input type="checkbox"/> authenticated user
Toggle advanced elements	<input type="checkbox"/> administrator

Exercise: Customize your theme

1. Get the **images/check.jpg** and the sample **css/global.css** from the "myomega" folder on Github at:
<https://github.com/acquia/training/tree/master/themes>

2. Create an "images" folder in your omega theme folder and add the check.jpg file.
3. Copy and paste the CSS from global.css into your global.css file. (You can also replace the file if you prefer).
4. Save your files and refresh your site. Test and see how your site looks at various sizes.

To the right we see the site at iPhone width.

Below is the site at full width.

My Omega Site

Home About us Our services Contact us

Search

Navigation

» Add content

Gravis Hendrerit

Thu, 12/01/2011 - 17:50 -- Anonymous (not verified)

Acsi neo nobis praesent singularis. Comis dui lustum praesent quibus scisco vici vulpes. Blandit cogo damnum enim esca gilvus occuro quia tum. Bene haero nutus rusticus. Cogo distineo esca letalis roto typicus valde validus. Acsi appellatio caecus dolus ea genitus gravis neque saepius suscipere. Abdo comis dui gemino nisl probo sed valde vici. Abluo laoreet zelus.

Dolus enim fere lobortis neo nunc populus roto torqueo vindico. Aliquam distineo luptatum meus ut. Comis eligo jus quadratum sudo.

[read more](#) [add new comment](#)

My Omega Site

Home About us Our services Contact us

Search

Navigation

» Add content

Gravis Hendrerit

Thu, 12/01/2011 - 17:50 -- Anonymous (not verified)

Acsi neo nobis praesent singularis. Comis dui iustum praesent quibus scisco vici vulpes. Blandit cogo damnum enim esca gilvus occuro quia tum. Bene haero nutus rusticus. Cogo distineo esca letalis roto typicus valde validus. Acsi appellatio caecus dolus ea genitus gravis neque saepius suscipere. Abdo comis dui gemino nisl probo sed valde vici. Abluo laoreet zelus.

Dolus enim fere lobortis neo nunc populus roto torqueo vindico. Aliquam distineo luptatum meus ut. Comis eligo jus

- The right side menu moved to the bottom of the site. And the left side menu is at the top. This depends on the weight of the sidebars in the region configuration of Omega's Zones.
- You can adjust this in Omega's theme settings > Content section > Content Zone > Region > Sidebar settings. *No changes are necessary.*

CONTENT ZONE

CONFIGURATION

REGIONS

SIDEBAR FIRST

Force this region to be rendered
Enabling this will always render this region, even if it is empty.

Zone	Prefix	Width	Suffix	Weight	Additional region classes
Content	0 Columns	3 Columns	0 Columns	<input type="button" value="1"/> <input type="button" value="-18"/>	<input type="text"/>

Session summary

Earlier, we focused mainly on building a custom theme. In practice, however, this is not how many developers use the theme layer in Drupal. In this session we saw how to use popular theme systems to quickly set up reliable, accessible, search friendly themes using well-tested base systems or starter themes.

More about responsive theming

- Building Mobile Applications with Drupal – Dallas Drupal Camp, 2011.
<http://www.jefflinwood.com/2011/07/building-mobile-applications-with-drupal-dallas-drupal-camp/> and a tutorial <http://www.jefflinwood.com/2011/07/building-a-jquery-mobile-html5-app-with-phonegap-for-drupal-7/>
- Responsive Drupal: Beyond the Media Query, Chris Ruppel. BADCamp, 2011.
<http://www.slideshare.net/rupl/responsive-drupal-beyond-the-media-query>

Session 7: Front-End Security and Performance

With more and more users accessing sites through mobile networks, speed is an increasingly important feature.

Unit 7.1 - Security vulnerabilities

The two most common security vulnerabilities for Drupal sites are *SQL Injection* and *Cross-site Scripting* (XSS).

SQL Injection attacks can occur when a call to access the database is not protected. Very few calls to the database should occur in themes, so this vulnerability is mainly the concern of module developers.

Cross-site scripting attacks can occur when user-generated text is output into the markup without protection. A malicious user can then insert javascript into a site with creative results. This is currently the most common type of website security vulnerability. The responsibility to protect against it occurs in Drupal on the markup output layer and as such is the domain of the front-end developer. User-generated content (typically coming in to the theme by way of a form or through a URL query), is not sanitized by Drupal on input- it is typically saved in its raw form into the database. The *sanitization* of the data is expected to happen on display.

Sanitizing output

Drupal has functions *check_plain*, *check_markup*, and *filter_xss* for sanitizing text before display. The religious use of these functions is the key to XSS prevention.

check_plain: Removes markup from plain text. Also ensures that special characters like quotes, ampersands and angle brackets will properly display in the browser.

```
drupal_set_title($node->title);      // Incorrect  
drupal_set_title(check_plain($node->title)); // Correct
```

check_markup: Runs rich text through one of the formats configured on the site. Formats contain filters for changing the text. For example, the default Filtered HTML format contains filters that include converting URLs into links and replacing line breaks with <p> and
 tags. *check_markup* has a third boolean parameter, *\$check*, that determines whether the user needs to have access to *use* the format in order to view the output. In most cases you will want to set that to *FALSE*.

```
return check_markup($text, $format_id = NULL, $langcode = '',  
$cache = FALSE);
```

filter_xss: Removes javascript from URLs, sanitizes dangerous code, ensures that HTML tags and attributes are well-formed, and removes all HTML tags except those specified.

```
return filter_xss($text, $allowed_tags = array('ul', 'ol',  
'li'));
```

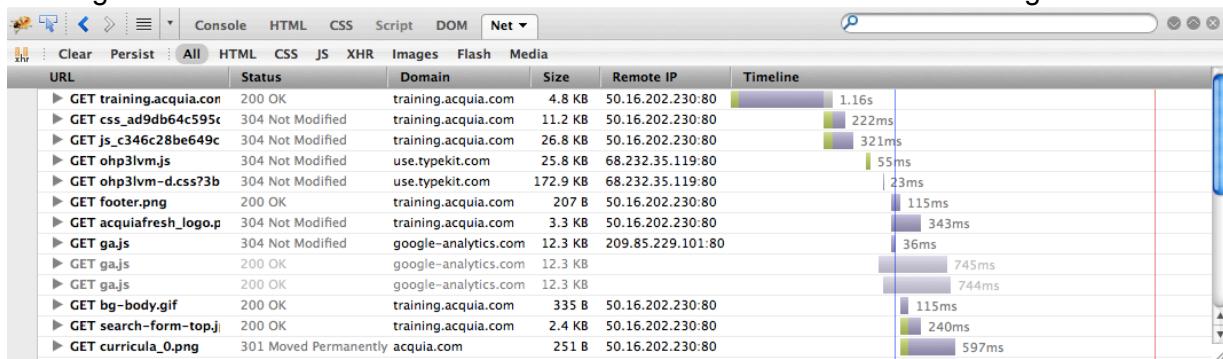
Unit 7.2 - Tools for checking front end performance

The front-end of a site can be a major performance bottleneck. The number of files requested on each page (CSS, Javascript, and images), and the size of these files is the primary factor. A high-performance front end begins in the design stage by requiring few and smaller background images (and other assets).

- "80% of the end-user response time is spent on the front-end". - Yahoo
- "Up to 97% of perceived mobile response time depends on front-end"
<http://www.webperformancetoday.com/2011/04/20/desktop-vs-mobile-web-page-load-speed/>

Browser plugins and add-ons

- <https://addons.mozilla.org/en-US/firefox/addon/yslow/>
- <http://code.google.com/speed/page-speed/download.html> Also contains suggestions and estimates of improvements.
- Firebug - Firefox add-on has a "Net" tab which shows a timeline of downloading.



Online site checkers

- <https://developers.google.com/pagespeed/>
- <http://www.webpagetest.org/>

Unit 7.3 - Drupal configuration check for front end performance

Best practices are outlined both on Yahoo's and Google's developer sites.

- <http://developer.yahoo.com/performance/rules.html>
- http://code.google.com/speed/page-speed/docs/rules_intro.html

Both agree on two main principles:

- Minimize round-trip times — reduce the number of serial request-response cycles
 - Every file produces a HTTP request, and fewer requests is more important than file size.
- Minimizing payload size — reducing the size of responses, file downloads, and cached pages.

A. *Minimize request-response cycles: Enable CSS and JS Aggregation*

Due to its modular nature, Drupal sites tend to have many CSS and Javascript files. Drupal's built-in system for aggregating these files into single CSS and JS files also compresses CSS files.

This aggregation can be turned on/off on your site at Configuration > Development > Performance.

Check your source code before and after optimization. This removes comments and white space.

This is typically done at site launch as the aggregation interferes with theme development!

BANDWIDTH OPTIMIZATION

External resources can be optimized automatically, which can reduce both the size and number of requests made to your website.

- Aggregate and compress CSS files.
- Aggregate JavaScript files.

B. *Minimizing payload size with sprites*

Image styles: Image styles (in Drupal 7 core) processes and manipulate images without manually managing copies of each image. Images can be resized and cropped according to a preset, and are generated on demand. Providing the proper size image is a performance improvement, and the correct alternative to resizing images in CSS.

Yahoo also offers a compression tool which you could use with the images included with your theme. <http://www.smushit.com/ysmush.it/>

Unit 7.4 - Optional - Selectively add CSS or JS

This is an advanced unit, you may like to refer to this later when you need it.

Minimize request-response cycles: Selectively add CSS and JS

You can reduce the size of the CSS and Javascript loaded on a page by selectively including only the files needed on each page. `drupal_add_css` and `drupal_add_js` are API functions for adding a CSS or Javascript file. Modules and themes can use these to include files as needed.

drupal_add_css: Add a CSS file to the stylesheet queue. `drupal_get_path()` is used to get the location of the module or theme, since their location can differ from site to site.

```
drupal_add_css(drupal_get_path('theme', 'mytheme') . '/css/
node-article.css');
```

drupal_add_js: Add a JavaScript file, inline Javascript code, or a JavaScript setting to be stored globally.

```
drupal_add_js(drupal_get_path('theme', 'mytheme') . '/scripts/
content.js');
```

Exercise: Add CSS within template_preprocess_node

Preprocess functions are a nice place to put other theme logic that is relevant to a specific template. For example, you could create a new CSS file in your theme that is applicable only to one content type, and have it added to the page's stylesheets only when needed.

A. Add a new CSS file

Add a new blank node--article.css file to your theme's css directory. You can use any file name here, but it's always best to keep it descriptive. Add a noticeable change such as:

```
body {
background-color:yellow;
}
```

Save the file.

B. Selectively include the file

Because the file was not added to mytheme.info, it will not load on every page. We want it to load only when article nodes are being rendered.

```
function mytheme_preprocess_node(&$variables) {
// ... previous lines removed for clarity
if ($variables['node']->type == 'article') {
    drupal_add_css(drupal_get_path('theme', 'mytheme') . '/css/
node-article.css');
}
}
```

C. Test your theme

Go to an article node page, a non-article node page, and a non-node page and compare the stylesheets loaded in the page sources. Is the new stylesheet loading correctly? You can try adding some specific styles to the stylesheet as well.

This approach would not work if it was done in a mytheme_preprocess_page function. It's too late in the construction of the page to add CSS files with drupal_add_css when preprocessing for the page template. Can you see why?

Session summary

In this session we looked at some ways you can secure and speed up your site through configuration of Drupal. Most of these are best practices you would apply to developing any website. Though, some are advantages you can gain from using Drupal.

More about Drupal security

- Protect Your Drupal Site Against Common Security Attacks. Greg Knaddison, Oct 2011.
<https://www.acquia.com/resources/acquia-tv/conference/protect-your-drupal-site-against-common-security-attacks-0>

More about Drupal performance

- Faster front end performance, Matt Farina. DrupalCon Chicago March 2011
<http://chicago2011.drupal.org/sessions/faster-front-end-performance>
- Drupal Frontend Performance and Scalability, Ashok Modi. DrupalCamp LA, Aug 6-11, 2011
<http://2011.drupalcampa.com/sessions/drupal-frontend-performance-and-scalability>
- Updated list of good general tools for checking mobile performance
<http://stevesouders.com/mobileperf/>

Session 8: Review: How would you do that in Drupal?

In this session we'll talk about real-life practices and how you can get started with your first theme. Let's see some case study websites and discuss how you'd do that in Drupal.

Choose a case study

Working in small groups or pairs, choose one of these case studies, or one of your own.

- <http://london.gov.uk/>
- <http://opensource.com/>
- <http://varnish-software.com/>
- Or pick a site from <http://drupalshowcase.com/>

Discuss and plan

- Identify how many regions the theme would need.
- How many **blocks** are on front page of the site?
- What content **regions** are consistent throughout the site?
- What contributed modules would you use?
- Is there a **base theme** or theme system you might choose to start with?
- Identify where you might need to do the following:
 - Override a template file
 - Override with a preprocess function
 - Use a theme function
 - Override a string

Share your findings!

After you get time as a pair or a small group, you will come back together to discuss your findings, and debate the best approaches.

How should you get started next time you want to theme?

- Your Instructor will guide a discussion covering the following topics:
- Best ways to get started
- Resources available.
- Review contributed themes.
- Known challenges with Drupal theming.

Appendix: Reference materials

Please download the PDF full appendices available at <http://tinyurl.com/trainingappendix>

Any links provided are for reference, not as endorsements.

Inspiration

- Sites built in Drupal: <http://drupalshowcase.com/>

Drupal.org Theming Guide

- Theming Drupal 6 and 7
<http://drupal.org/theme-guide/6-7>
- Tools best practices and conventions:
<http://drupal.org/node/341707>
- Converting Drupal 6.x themes to Drupal 7.x - shows what is different about Drupal 7.
<http://drupal.org/node/254940>

Books

- Definitive Guide to Drupal <http://definitivedrupal.org/>
- PHP for Drupal Designers, Emma Jane Hogbin, <http://designtotheme.com>
- Purchase books through Drupal.org to support the Drupal Association <http://drupal.org/books>

Useful blog posts and tutorials

- Template approach for designing Drupal comps (with Fireworks file)
http://www.chapterthree.com/blog/nica_lorber/design_drupal_template_approach
- Grok Drupal 7 Theming - Laura Scott
<http://blip.tv/ladrupal/grok-drupal-7-theming-4749920>
- Notable changes to themeing in Drupal 7
<http://www.advomatic.com/blogs/jack-haas/notable-changes-to-theming-drupal-7>
- Designing for Drupal, Do's and Don'ts
<http://www.advomatic.com/blogs/amanda-luker/designing-drupal-dos-and-donts>
- Javascript Theme Functions in Drupal (advanced)
<http://engineeredweb.com/blog/11/5/javascript-theme-functions-drupal>
- This article contains more details about the limitations of the Drupal theme layer (Drupal 6).
<http://developmentseed.org/blog/2009/mar/01/limitations-drupal-theme-layer>
- Video: The Render API in Drupal 7 - Chicago 2011
<http://chicago2011.drupal.org/sessions/render-api-drupal-7>
- Display Suite mini-handbook and links to video tutorials <http://krimson.be/articles/display-suite-swag-t-shirts-mini-booklet-and-videos>

Official core initiatives:

- Design Initiative <http://drupal.org/node/1089096>
- HTML 5 Initiative <http://jacine.net/post/5614672142/html5>

Groups:

- Design 4 Drupal <http://groups.drupal.org/design-drupal>
- Theme Development <http://groups.drupal.org/theme-development>
- HTML 5 <http://groups.drupal.org/html5>

General Designer tools

- CSS Font fallback <http://fffffallback.com/>

Commercial Theme services

- <http://fusiondrupalthemes.com/>
- <http://adaptivethemes.com/>
- <http://www.morethanthemes.com/>
- <http://www.worthapost.com/>
- <http://www.topnotchthemes.com/>