DrupalCamp CT 2012

Intensive Beginning Theming

Preston So

August 18, 2012

Welcome!

- **Preston So** (prestonso) is Prototyper Intern at Acquia and co-maintainer of the upcoming Spark distribution. He founded the Southern Colorado User Group.

  www.prestonsodesign.com
  drupal.org/user/325491
  pso@college.harvard.edu
  preston.so@acquia.com

What we'll cover

1. The Drupal theme layer
2. Theme structure and creation
3. Template files: page.tpl.php
4. Theming content types and views
5. Theme preprocessing
   (if time permits)

Some diagnostics

- Have you ever ...
  - used Drupal?
  - installed and configured Drupal?
  - installed and configured a Drupal theme?
  - written HTML and CSS?
  - edited files provided in default themes?
  - used a base or starter theme?

Local Drupal

- For these tasks, we'll need a local installation of Drupal to serve as a sandbox.
- If you already have an AMP stack, download and install Drupal.
- Acquia Dev Desktop is a great tool to use:

https://www.acquia.com/downloads

Getting a head-start

- To code alongside us, download this .zip to get started with a bare-bones starter kit.

dl.dropbox.com/u/34693193/prestomaticlive.zip

- To read the completed theme in all its glory, download this .zip for the whole shebang.

dl.dropbox.com/u/34693193/prestomatic.zip

Let's get started!

- Tools
  - A good text-editor you're comfortable with:
    - Free options: Notepad++ for PC or Textwrangler for Mac OS
    - Or whatever you prefer.
- Theme Developer module
  - drupal.org/project/devel_themer
  - drupal.org/project/devel

Let's get started!

- An in-browser developer tool
  - Firebug for Firefox
  - Inspect Element for Chrome
- A good attitude
  - Learning is fun, but it can also be frustrating; don't fret, we have beach balls

The Drupal theme layer

1

Why theme?

What is a theme?

How does Drupal handle themes?

What is a theme?

- Markup
  - A theme contains markup of your site's layout and structure
- Styles
  - A theme also contains the styles that make up your site's look and feel
- Fun stuff
  - Themes can also contain images and JavaScript

What makes a Drupal theme different?

- Drupal assumes nothing
  - Drupal is a framework, you decide how it's built. This makes it a little less "out-of-the-box" then say WordPress.
- Flexibility
  - Drupal has multiple ways to solve problems; other CMSes usually have only a couple ways.
- Overrides
  - Overriding offers the advantages of being more extensible, you can theme specific modules, etc.

Why write a theme?

- One aspect of Drupal's powerful functionality is its versatility in handling almost any design.
- With a theme you can override any part of Drupal's *themable output*.
- How is Drupal theming different from other CMSes?

Themes give you

- Manipulability
  - Full control of the theme layer
- Extendability
  - You can go from lean and mean to crazy ridiculous
- Efficiency
  - No cruft when starting from scratch

Good starter themes

- Zen
  - A nice clean starting point, well-coded
  - drupal.org/project/zen
- Framework
  - In between Zen and Stark, based on a 960 grid
  - drupal.org/project/framework
- Stark
  - Completely bare-bones, but a nice starter
  - drupal.org/project/stark

Theming vocabulary

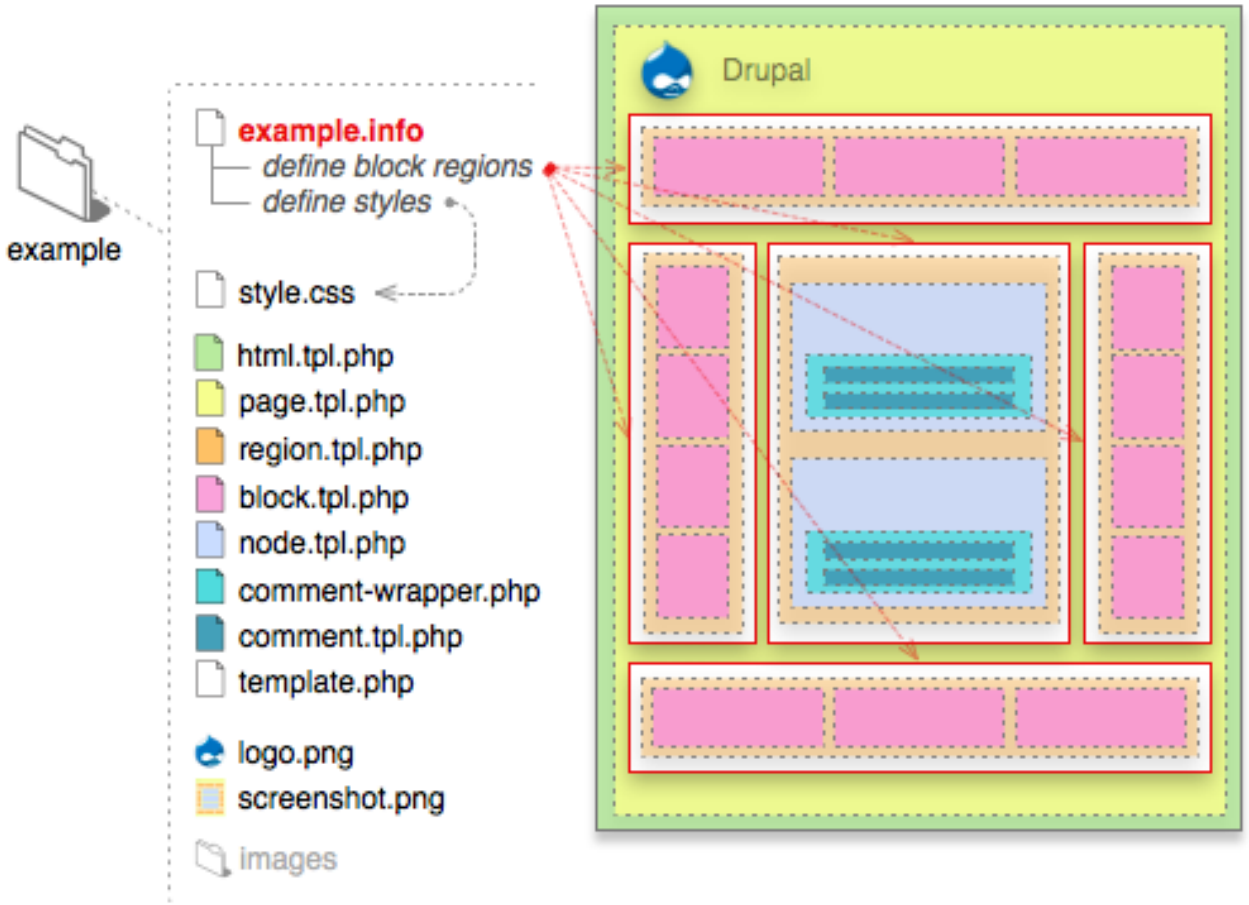
- Node
  - Most simply, a unit of content, e.g. page, article, post.
- Block
  - Output repeatable across pages, e.g. most viewed, last updated, user login boxes.
- Region
  - Contains themable output, e.g. sidebar, footer.
  - Contains blocks.

Theming vocabulary

- Page
  - Contains regions. This is what the user sees.
- Template
  - Organizes the themable output.
- Sub-theme
  - Inherits resources from the parent theme.
  - Located in a subdirectory.

The Drupal theme layer

- The theme consists of a collection of files.



Theme structure and creation

2

How do I create a theme?

What does a theme need?

The structure of a theme

- Themes are contained within a directory carrying that theme's name, e.g. bartik, garland.
  - Core themes are located in /themes/.
  - Contributed themes (such as the one we are creating) are located in /sites/all/themes/.
- To tell Drupal about a new theme, all you need to do is insert a .info file inside that directory:

/sites/all/themes/prestomatic/prestomatic.info

Theme files

- prestomatic.info
  - The theme settings.
- style.css
  - Our handy dandy style sheet.
- template.php
  - Handles overrides of module and system output.
  - We'll get here, time permitting.

Template files

- node.tpl.php
  - Template for all nodes.
- block.tpl.php
  - Template for all blocks.
- page.tpl.php
  - Template for all pages.
  - This is where most layout happens.

The .info file

- The .info files contains settings and configuration information for each theme.
- Regions based on your layout, to be shown on the blocks configuration page, are set here.
- The only required keys are *name*, *core*, and in most cases *engine*.

.info defaults

- Defaults for .info files can be found on d.o:

  http://drupal.org/node/171206

- Defaults only apply when the key is absent from the .info file. For example, *region* defaults will only apply if there is no *region* in .info.

.info keys

- name
  - Our theme's human-readable name.
- description
  - Our theme's ticket out of nondescriptness.
- screenshot
  - Optionally, declare a screenshot to be shown on the theme configuration page.

.info keys

- core
  - Which version of Drupal is this for?
- engine
  - What template engine does the theme use?
  - Usually *phptemplate*, the default value.
- regions
  - Identify the regions that comprise the layout.

.info keys

- features
  - Page elements output out of the box by the theme that can be enabled or disabled.
  - Creates checkboxes in theme configuration.
- stylesheets
  - CSS files.
- scripts
  - JS files.

prestomatic.info

- First, let's give our theme an identity.

```
name = Prestomatic
description = Example theme for D4D4
```

prestomatic.info

- Now, we tell Drupal some essentials.

```
name = Prestomatic
description = Example theme for D4D4
core = 7.x
engine = phptemplate
stylesheets[all][] = style.css
```

prestomatic.info

- Let's make this theme responsive ... sort of:

```
name = Prestomatic
description = Example theme for D4D4
core = 7.x
engine = phptemplate
stylesheets[all][] = style.css
stylesheets[{max-width:480px}][] = 480.css
```

prestomatic.info

- Let's insert some regions based on our layout.

```
name = Prestomatic
description = Example theme for D4D4
core = 7.x
engine = phptemplate
stylesheets[all][] = style.css
stylesheets[{max-width:480px}][] = 480.css
regions[header] = Header
regions[navigation] = Navigation
```

prestomatic.info

- Brackets surround machine names.

```
regions[header] = Header
regions[navigation] = Navigation
regions[highlighted] = Highlighted
regions[help] = Help
```

prestomatic.info

- The *content* region is required.

```
regions[header] = Header
regions[navigation] = Navigation
regions[highlighted] = Highlighted
regions[help] = Help
regions[content] = Content
regions[sidebar_first] = First sidebar
regions[sidebar_second] = Second sidebar
regions[footer] = Footer
```

Clearing the cache

- Whenever you add new files to Drupal's directory structure, you should clear the cache.
- Navigate to *Configuration* > *Development* > Performance and click *Clear all caches*.
- Hooray! We've just created a theme.

Template files: page.tpl.php

3

How do I place elements?

What variables do I call?

How do I condition display?

The .tpl.php

- Template files (.tpl.php, or *tipple-fips*) are the means through which Drupal spits out the HTML for the final end user experience.
- There are limitless possibilities of template files, but we will only focus on a few for now.
- Template files are written in PHP, but fear not! Theme PHP is simple and readable.

page.tpl.php

- The *page.tpl.php* file controls all of the regions (containing blocks) and major sidewide variables.
- This template file is where the most fundamental layouts are determined.

page.tpl.php

- Those following along in their code can open page.tpl.php to see a rough HTML structure.

page.tpl.php

- Let's add in some regions to start off: a very easy *print* command. In line ######:

```
<header id="header" role="banner"
    class="clearfix">
  <?php print render($page['header']); ?>
  <nav id="navigation" role="navigation"
    class="clearfix">
```

page.tpl.php

- Add the following regions to your *page.tpl.php*. Recall that PHP needs machine-readable names.
  - *Header*
  - *Highlighted*, place inside #highlighted
  - *Help*, place after Highlighted
  - *Content,* place after Help
  - *Sidebar first*, place inside #sidebar-first
  - *Sidebar second,* place inside #sidebar-second
  - *Footer*, place inside #footer

page.tpl.php

- Let's add in some important site information variables too.
  - *$site_name*, the name of the site
  - *$site_slogan*, the slogan, if enabled
  - *$breadcrumb*, breadcrumb links
  - *$messages*, Drupal's messages about site status
  - *$title*, the title of the page
    - *$title_prefix* and *$title_suffix*, accompaniments to the title
  - *$tabs*, Drupal's tabs on each page

Conditions in page.tpl.php

- What if we want to hide the surrounding HTML structure when there is nothing to insert?
- For example, do we want the *h2* for site slogan appearing even if there is no site slogan?
- More significantly, do we want the *divs* surrounding the sidebar regions if the regions might be empty on certain pages?

page.tpl.php

- With PHP conditions, we can display the HTML only if there's stuff inside. Take *$site_slogan*.

```
<?php if ($site_slogan): ?>
  <h2 id="site-slogan">
    <?php print $site_slogan); ?>
  </h2>
<?php endif; ?>
```

page.tpl.php

- Display *$site_slogan* only if *$site_slogan* is actually defined.

```
<?php if ($site_slogan): ?>
  <h2 id="site-slogan">
    <?php print $site_slogan); ?>
  </h2>
<?php endif; ?>
```

Logical operators in PHP

Operators	Name	Description
x and y x && y	And	True if: both x and y are true
x or y x || y	Or	True if: either or both x and y are true
x xor y	Xor	True if: either x or y is true, but not both
! x	Not	True if: x is not true

page.tpl.php

- Display *$site_slogan* only if *$site_slogan* AND *$site_name* are defined.

```
<?php if ($site_slogan && $site_name): ?>
  <h2 id="site-slogan">
    <?php print $site_slogan); ?>
  </h2>
<?php endif; ?>
```

Conditions in page.tpl.php

- We can expand this logic to many different places, including layout regions. Attach conditions to the following:
  - *$site_slogan*, child of h2#site-slogan
  - *$highlighted*, child of div#highlighted
  - *$title*, child of h1#title
  - *$tabs*, child of div#tabs-wrapper
  - *$sidebar_first*, child of aside#sidebar-first
  - *$sidebar_second*, child of aside#sidebar-second

block.tpl.php and node.tpl.php

- Just as *page.tpl.php* is the template file that governs how overall pages are displayed,
- *block.tpl.php* is the template file that governs how blocks are displayed (in regions).
- Similarly, *node.tpl.php* is the template file that governs how nodes are displayed in the full node view form.

html.tpl.php

- You'll notice that there is no *doctype* declaration or *head* in the *page.tpl.php* file.
- This is because *html.tpl.php* controls the *head*. We want Drupal to do this, since we declared styles and scripts in *prestomatic.info*.
- This is also where body classes are declared.

So many variables!

- All template files contain a multitude of PHP variables that are also called in core templates.
- You can find the core template files at d.o:

Core templates and variables

- All template files contain a multitude of PHP variables that are also called in core templates.
- You can find the core template files at d.o:

http://drupal.org/node/190815

Theming content types and views

4

How do I theme content types?

How do I theme views?

How do I find out what I can theme?

Specificity

- We can get even more specific with our template files. These files override page.tpl.php.

page--node--1.tpl.php
page--node.tpl.php
page--front.tpl.php
page.tpl.php
system's page.tpl.php

page--front.tpl.php

- We can use *page--front.tpl.php* to create a template that is responsible for what displays on the home page, or <*front*>.
- This is particularly useful for sites that have a different design for the front page and the rest of the site.

Even more template files

- We have content types *article*, *basic page*, and perhaps other user-defined ones, like *photo*.
- What are these template files refer to?

page--article.tpl.php
page--photo.tpl.php

page.tpl.php
(Basic page)

Even more template files

- What are these template files refer to?

node--article.tpl.php
node--photo.tpl.php

node.tpl.php
(Basic underlying node)

Even more template files

- What are these template files refer to?

node--article.tpl.php
node--photo.tpl.php

node.tpl.php
(Basic underlying node)

Eureka!

- There is no limit to the number of template files you can have in your theme. If you desire, you can create template files for each and every node available on the site.

- So themes are very powerful. Not only can we use template files for content types, we can use them for Views and other modules as well, extending our reach as themers indefinitely.

Theming views

- If you do not have Views installed, let's go ahead and put it on our Drupal install.

  drupal.org/project/ctools
  drupal.org/project/views

- Let's talk about how views are structured and the themable output the Views module presents us.

Theming views

- If you do not have Views installed, let's go ahead and put it on our Drupal install.

  drupal.org/project/ctools
  drupal.org/project/views

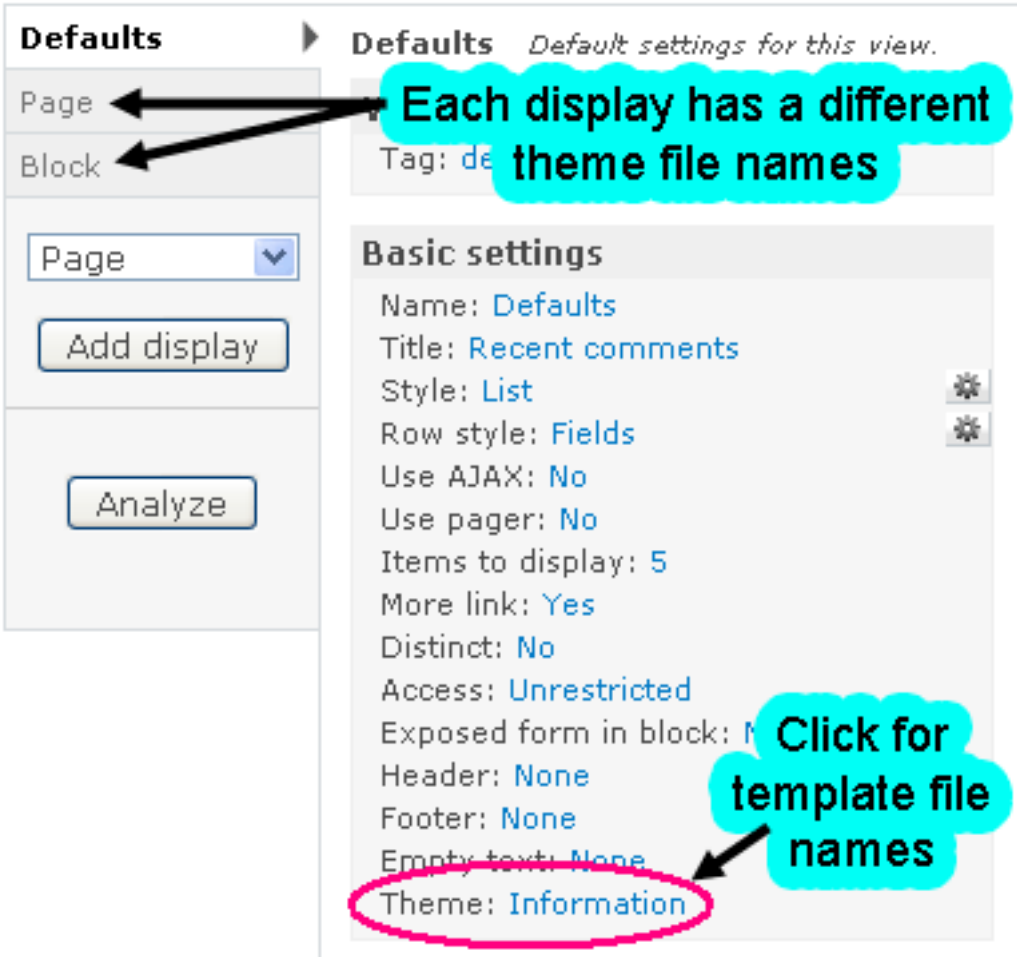
- Let's talk about how views are structured and the themable output the Views module presents us.

View structure

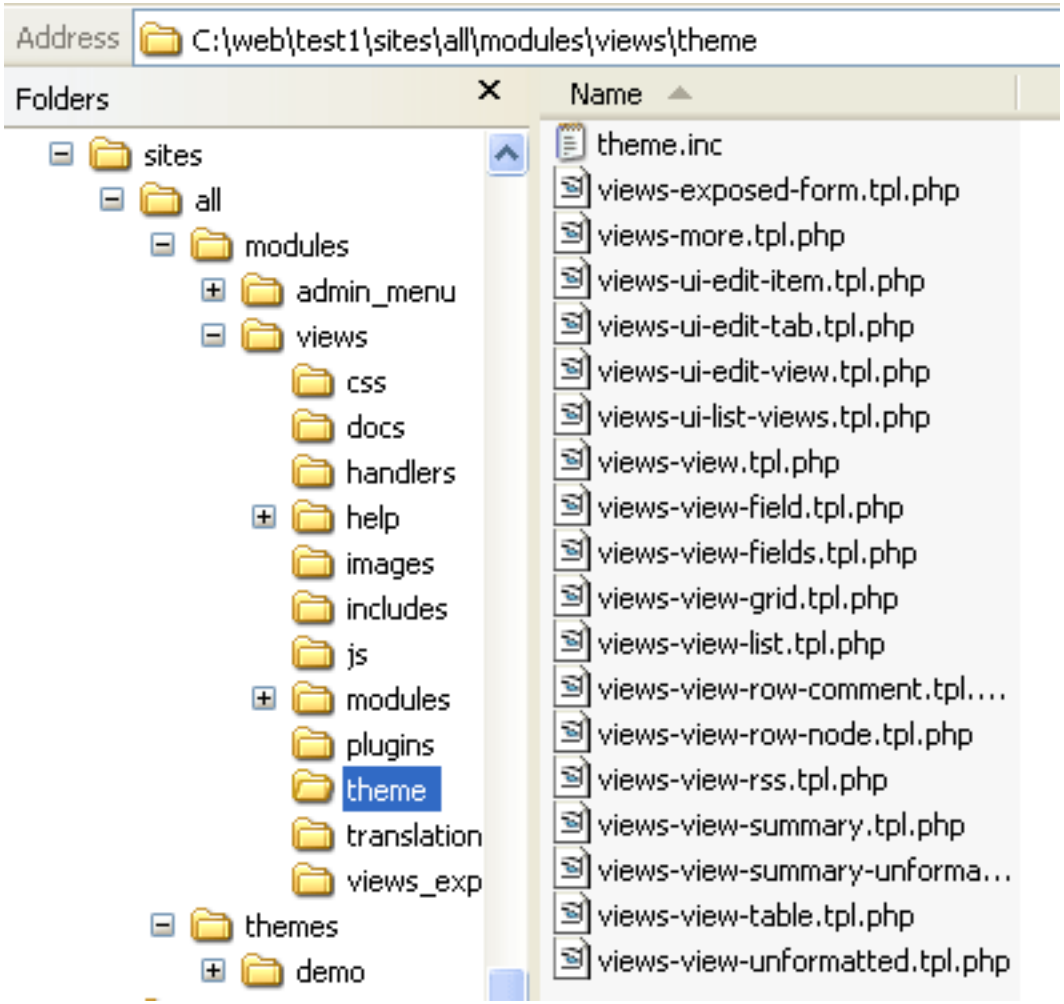
www.group42.ca/theming_views_2_the_basics

- Field
  - Individual field as displayed in a view.
- Row
  - Groupings of multiple fields.
- Style
  - Controls assembly of rows, e.g. *table*, *unordered list*.

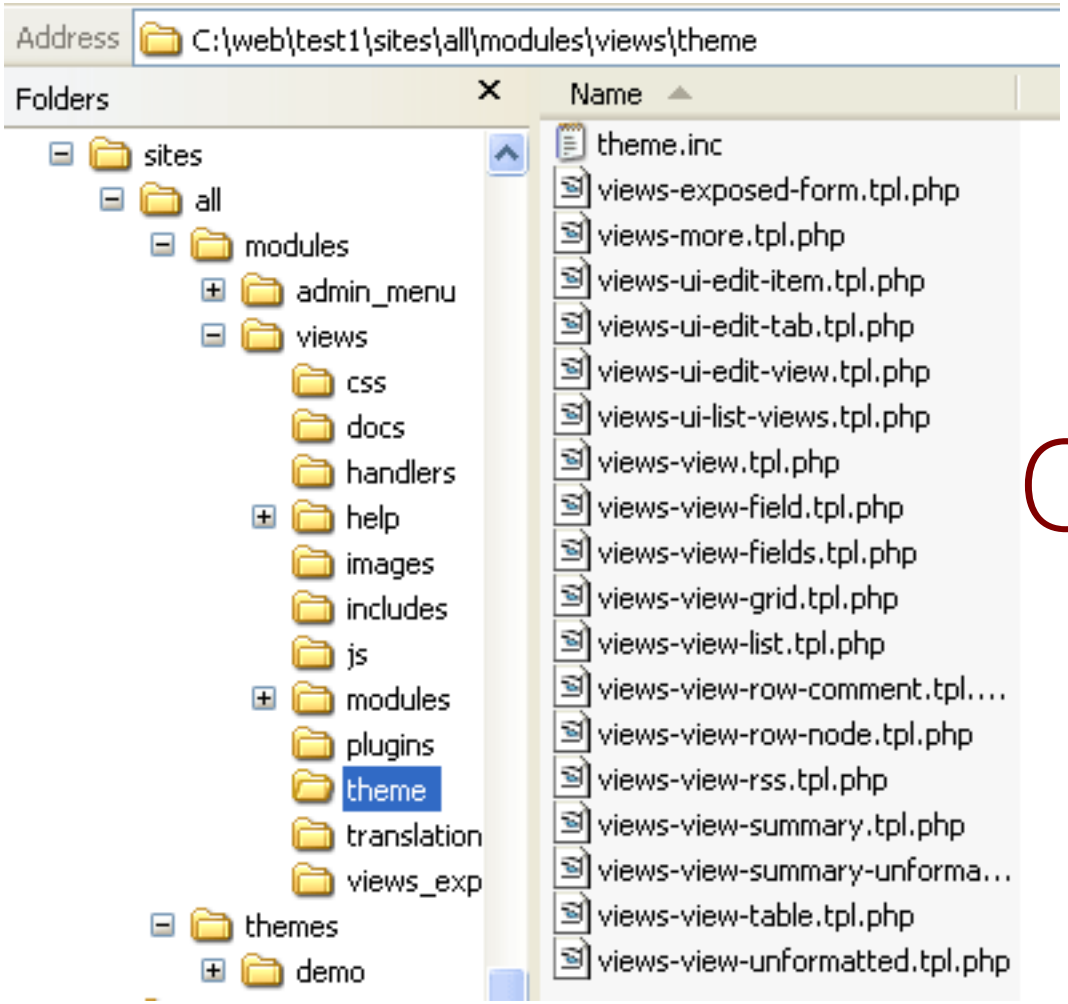
View template files



View template files



View template files



OMG

View template files

- Don’t worry, we only need the particular templates that we are using.
- views-view-field.tpl.php
  - Individual field.
- views-view-list.tpl.php
- views-view-table.tpl.php
  - The style for a particular set of fields.

More theming

- **Drupal theme guide**
  drupal.org/documentation/theme
- **Drupal theme API**
  drupal.org/node/722174
- **Design to Theme**
  www.designtotheme.com

Welcome!

- **Preston So** (prestonso) is Prototyper Intern at Acquia and co-maintainer of the upcoming Spark distribution. He founded the Southern Colorado User Group.

  www.prestonsodesign.com
  drupal.org/user/325491
  pso@college.harvard.edu
  preston.so@acquia.com