

DrupalCamp CT 2012

# Drupal-Aware Design: Techniques for Better Themes

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](http://www.prestonsodesign.com)  
[drupal.org/user/325491](http://drupal.org/user/325491)  
pso@college.harvard.edu  
preston.so@acquia.com



# What we'll cover

- 1 Drupal design principles
- 2 The “Drupal-aware” spectrum
- 3 Drupal-aware design
- 4 Drupal-aware theming
- 4 To Drupal and beyond

Where are  
we now?

# Where are we now?

- Design in Drupal has come a long way.

# The White House

the WHITE HOUSE PRESIDENT BARACK OBAMA

★★★★★ ★★★★★

 THE WHITE HOUSE WASHINGTON

[Get Email Updates](#) | [Contact Us](#)

 [BLOG](#) | [PHOTOS & VIDEO](#) | [BRIEFING ROOM](#) | [ISSUES](#) | [the ADMINISTRATION](#) | [the WHITE HOUSE](#) | [our GOVERNMENT](#)

## MIDDLE CLASS TAX CUTS

If Congress fails to act, a typical middle-class family of four will see its taxes go up by \$2,200, and America's small business owners would take a big hit. President Obama refuses to let that happen.

[Learn More](#)

1 2 3 4



President Barack calls on Congress to extend tax cuts for middle class families, White House Photo, Lawrence Jackson, 7/9/12

### FEATURED TOPICS



**The To Do List for Congress**  
Before Congress leaves for summer recess, President Obama is calling on them to take action to create jobs and help the middle class.

[◀ BACK](#) | [NEXT ▶](#)

**THE BLOG** 

**WHITE HOUSE SCHEDULE** 

### SEARCH the SITE

[Search](#)

**WE the PEOPLE** YOUR VOICE IN OUR GOVERNMENT  
Create and Sign Petitions on WhiteHouse.gov

# Pieces Inc.

# pieces

[Home](#)   [New arrivals](#)   [About](#)   [Questions](#)   [Press](#)   [Contact](#)

Google™ Custom Search

v Wool White Wall Decorations Throws Taupe Tables & Cabinets Stone Rugs Resin Red Rattan Porcelain Pink Pillows Pet Orange Natural Natural Mirror  
l Lucite Leather/Fur Lamps Headboards Grey Green Gold Glass Floor & Table Accessories Chrome Chandeliers Chairs & Sofas Ceramic Cane Brown  
Black



Get first notice when new items arrive.

Pieces 3234-A Roswell Road • Atlanta, GA 30305

P 404.869.2476 • F 404.869.2477 • [Email](#)

Follow us:

We'll be back later..

# Center for Investigative Reporting



CENTER FOR  
INVESTIGATIVE  
REPORTING

Revealing injustice for 35 years



## Re-greening the Sahel

Niger races to reclaim the desert through an innovative initiative that puts Nigeriens to work to help rebuild arable land and avoid famines during recurring droughts.

About California Watch The Bay Citizen

Email Alerts Twitter Facebook Tumblr

SUPPORT US ▾

Home

Projects

Blogs

Media ▾

Topics ▾

Search

## Projects



Notice to Appear



The Nuclear Zone



Food for 9 Billion

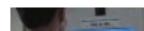
## LATEST REPORTS

Length  
- Any -

Topic  
- Any -

Media Type  
- Any -

Jul 19, 2012



Tech designed to 'read' body's

G.W. Schulz  
Project(s)

RT @thebaycitizen: Pentagon allows military members to march in uniform in San Diego's gay pride parade this weekend: <http://t.co/HJNQdfdW>

@CIRonline Jul 19, 2012 - 09:20 pm

# Sites these days

- About four years ago, Dries mentioned that 3 out of 4 Drupal sites are ugly.
- That's changed (pretty substantially), but what has replaced ugly Drupal designs? Ugly Drupal implementations of themes that are not extensible or flexible.

# The problem

- There is a dramatic change in process between design and theme.
  - *Structure.* Photoshop layers vs. theme templates.
  - *Goal.* Good design vs. integration with Drupal.
  - *Process.* Putting things together vs. making things display right.
  - *Next steps.* Focus on the theme vs. focus on the Drupal implementation.

# Weaknesses

- Handovers in markup cause consternation and disapproval when later stages are reviewed.
- Without a complete understanding of the design, the themer does not know how to approach the design.
- Without a complete understanding of the theming process, the designer does not know how to approach Drupal.

# Let's unify these approaches

- We can bring designers and themers together for a more robust design and theme process.
- Designers and themers need to know at least the basics of what each is doing.

# Design principles are useful

- Design principles are worthy sources of wisdom not only for the design process, but also for our project process.
- Where might some design principles come into play at the design-theme interface?

1

# Drupal design principles

Postel's robustness principle

General design principles

Drupal 7 UX principles

# Design principles

- Design principles by Jeremy Keith  
[principles.adactio.com](http://principles.adactio.com)
- Drupal 7 UX design principles  
[www.d7ux.org](http://www.d7ux.org)

# Postel's robustness principle

Be conservative in what you send;  
be liberal in what you accept.

- In the original sense, most simply, code sent to other places should follow the recipient's specifications, while code that is received can be handled even if it does not conform.

# Postel's robustness principle

Be conservative in what you send;  
be liberal in what you accept.

- Designers should ensure their designs transition gracefully into diverse code.
- Themers should write themes taking into account many different possibilities.

When browsers are lax about what they expect, the system works better but also it encourages laxness on the part of web page writers. The principle of tolerance does not blunt the need for a perfectly clear protocol specification which draws a precise distinction between a conformance and non-conformance. The principle of tolerance is no excuse for a product which contravenes a standard.

**TIM BERNERS-LEE**  
1998

# Drupal is our DNA

We follow the standards of Drupal.

Where standards are weak,  
we champion new standards and  
prove why those standards are  
more effective.

- When designing and implementing designs in Drupal, should we stick to Drupal paradigms across the board, even in HTML/CSS?

# Drupal is our DNA

The answer is sort of.

- Yes cases

The site will be in Drupal.

The theme will be contributed.

- Maybe cases

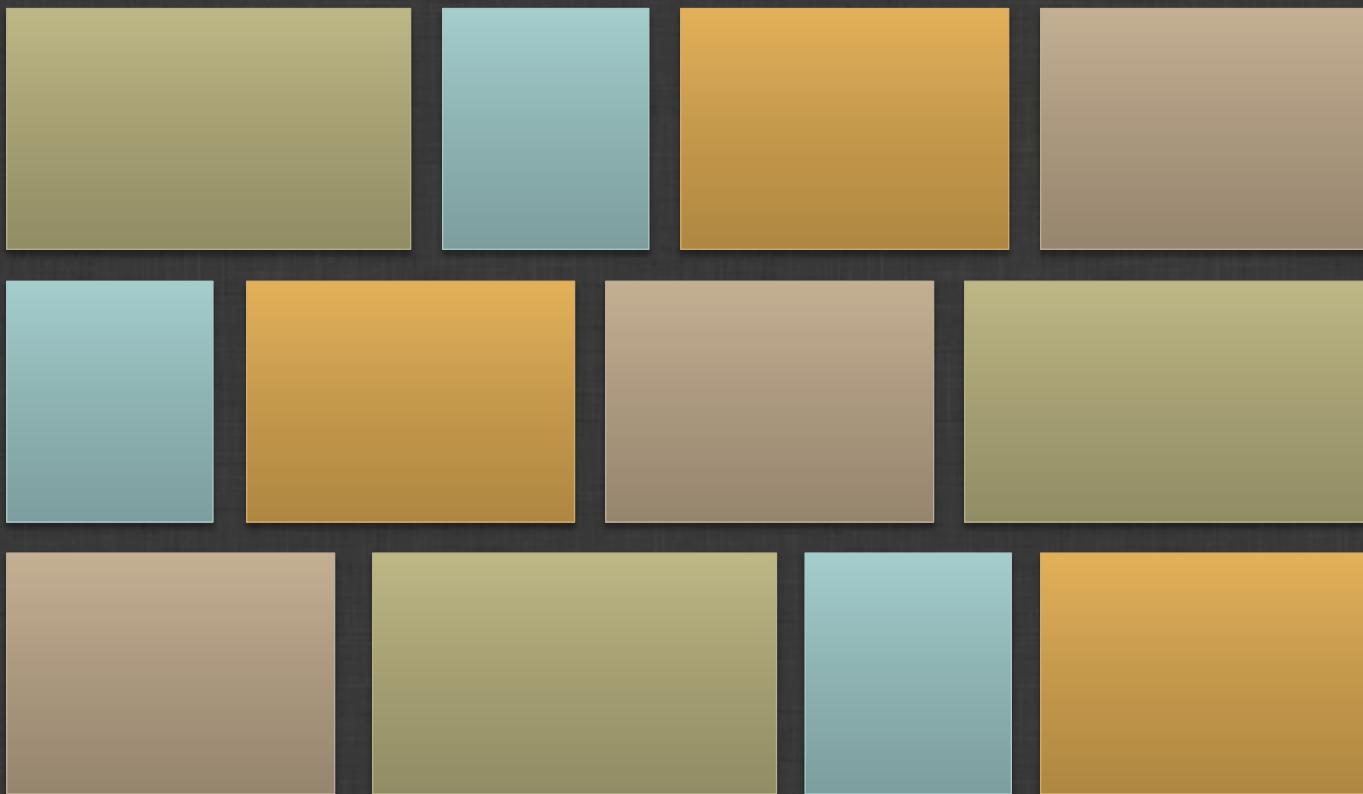
The site may not be in Drupal forever (e.g. prototype).

The theme may need to be Drupal-agnostic.

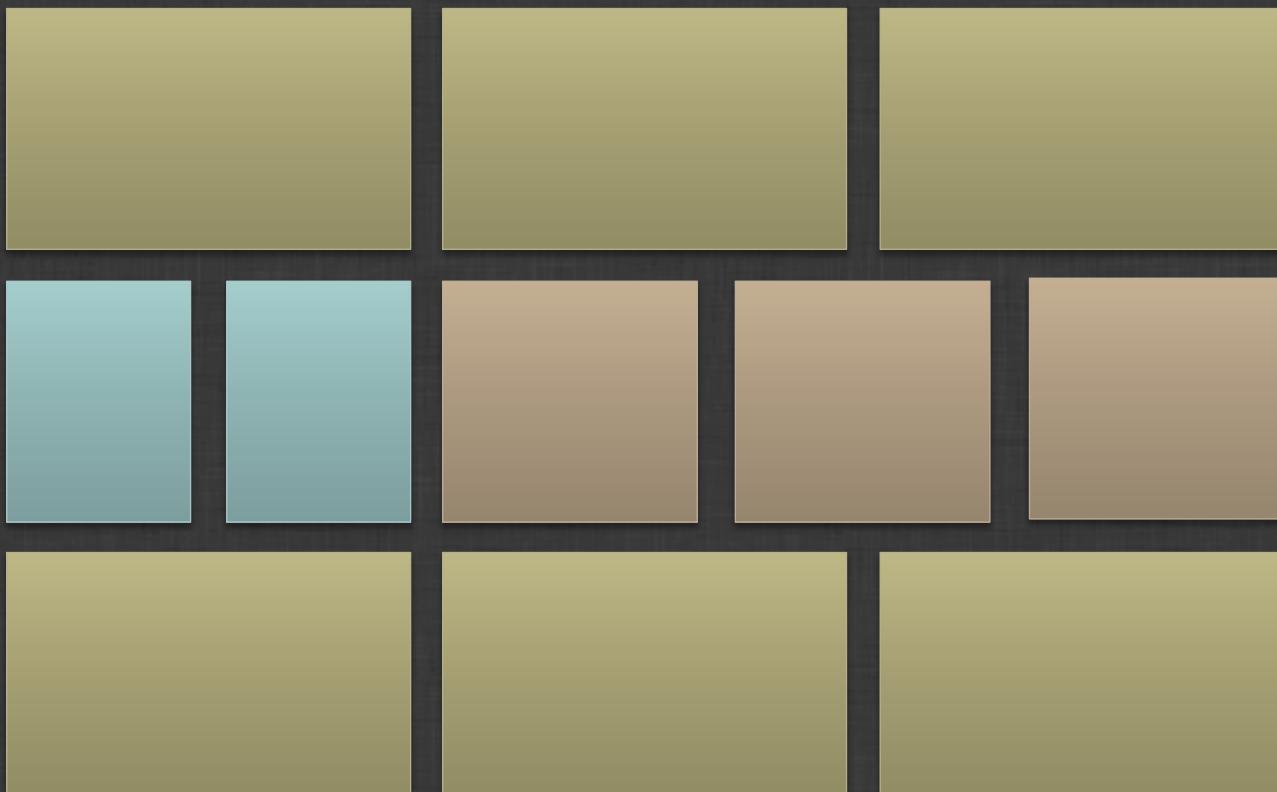
Good design is  
as little design  
as possible.

DIETER RAMS  
1976

# Efficient design



# Efficient design



# Drupal 7 UX project principles

Design for the 80%.

Privilege the content creator.

Make the default settings smart.

# The “Drupal-aware” spectrum

What is Drupal-aware?

From Drupal-agnostic ...

To embracing Drupal

# The “Drupal-aware” spectrum

- What is designing *for* Drupal?
  - Using standard Drupal areas and regions
  - Using particular Drupal classes (e.g. *page-front*)
  - This methodology embraces Drupal.

# The “Drupal-aware” spectrum

- What is designing *with* Drupal?

*Anticipating* standard Drupal areas and regions

*Anticipating* particular Drupal classes (e.g. *page-front*)

This methodology is *aware* of Drupal, or perhaps even *agnostic* to Drupal.

# The “Drupal-aware” spectrum

- If the use of Drupal is uncertain, we should tailor our code to anticipate all possibilities.
- What might this look like?

# The “Drupal-aware” spectrum

- Ask yourself some questions about your intent:
  - Will the theme be contributed?
  - Will the theme have sub-themes?
  - Will the design need to change from Drupal?
  - Will the theme need to change from Drupal?
- Then, tailor *both* the design and theming processes to this intent.

# The “Drupal-aware” spectrum

*away from Drupal →*

Drupal-only

Drupal-aware

Drupal-  
agnostic

# A Drupal-only process

*away from Drupal →*

Drupal-only

Drupal-aware

Drupal-agnostic

- Fast theming
- Low tolerance for non-Drupal markup

# A Drupal-only process

- Our design is built using Drupal classes/names.

```
ul.primary {  
    /* primary links CSS */  
}  
ul.secondary {  
    /* secondary links CSS */  
}
```

# A Drupal-agnostic process

*away from Drupal →*

Drupal-only

Drupal-aware

Drupal-  
agnostic

- Slower theming
- High tolerance for non-Drupal markup

# A Drupal-agnostic process

- We can use whatever markup we desire.

```
nav#primary {  
    /* primary links CSS */  
}  
  
nav#secondary {  
    /* secondary links CSS */  
}
```

# What is Drupal-aware?

- As we move closer to Drupal as a need on this spectrum, we should ensure that our designs and themes utilize classes and elements that are most conducive to Drupal.
- But a “Drupal-aware” theme should not only run well on Drupal but also other CMSes.
- This is a fusion of both Drupal-only and Drupal-agnostic approaches.

# What is Drupal-aware?

- A selection of one of these processes does not mean that it is the end-all, be-all solution.
- As with every project, approaches need to be analyzed and tested. No one solution will work for all conditions.
- This spectrum model is only intended to allow you to visualize the possible approaches to follow.

# What is Drupal-aware?

- A Drupal-aware methodology is a great idea for rapid prototyping, rapid implementation, or simply to test a variety of conditions.
- Jeff Noyes, “Rapid Prototyping with Drupal”,  
Design 4 Drupal 2009  
[www.youtube.com/watch?v=O25Wuts90sw](http://www.youtube.com/watch?v=O25Wuts90sw)

# A Drupal-aware process

*away from Drupal →*

Drupal-only

Drupal-aware

Drupal-  
agnostic

- Account-for-all theming
- Tolerance for both markup styles

# A Drupal-aware process

- Account for myriad possibilities in the CSS:

```
ul.primary, nav#primary {  
    /* primary links CSS */  
}  
  
ul.secondary, nav#secondary {  
    /* secondary links CSS */  
}
```

# The ideal Drupal-aware theme

- So what is the endgame in terms of perfecting the convergence of design and theming methodologies and processes?
- A design should be easily integrated with Drupal, but it should leave other options open.

# The ideal Drupal-aware theme

- A theme should be usable in any Drupal scenario, and it should most ideally lack extra, redundant code as well as presentational material.

# 3

## Drupal-aware design

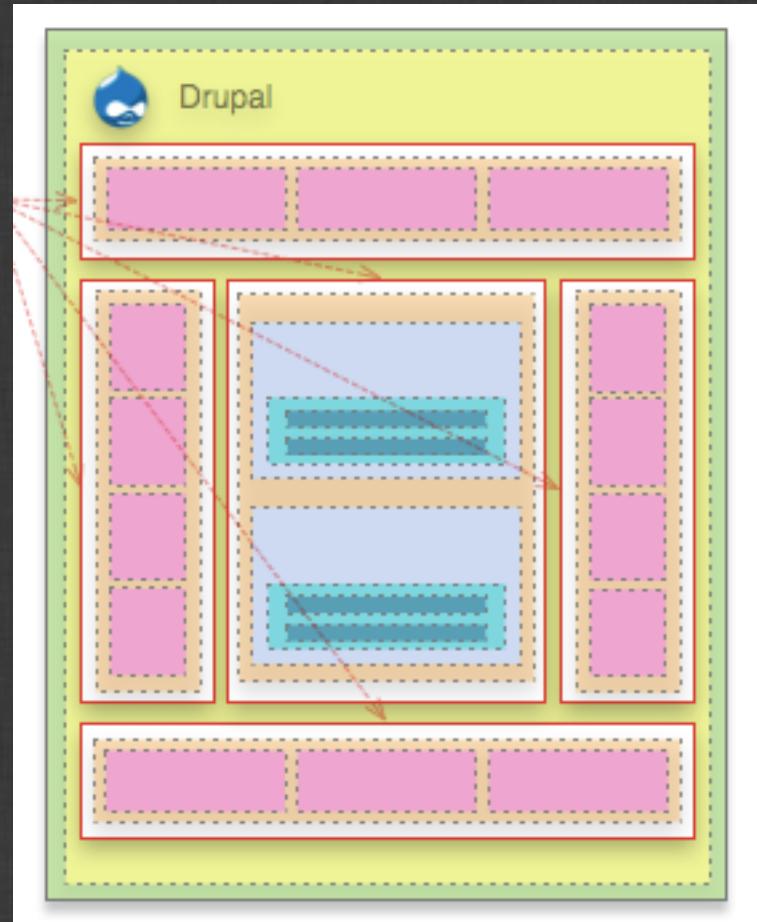
The stacked-paper model

Standards and conventions

Usability concerns and considerations

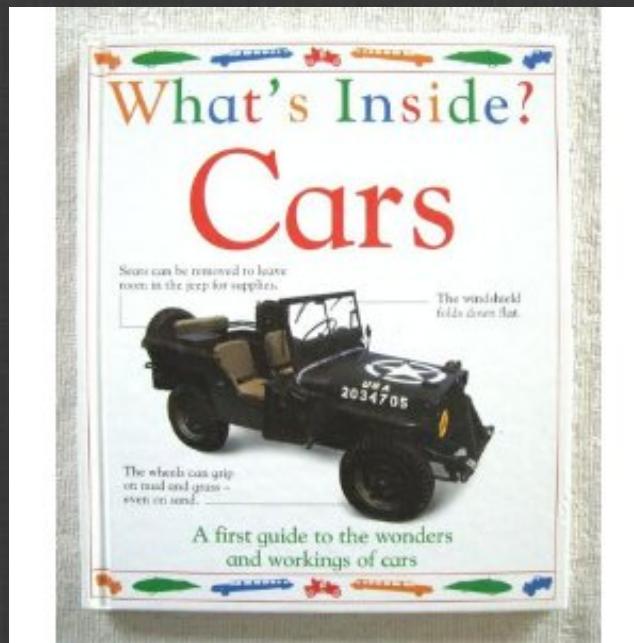
# For designers

- As usual, envision designs and wireframes using an approach based on the user experience.
- D7UX: Design for the 80%.



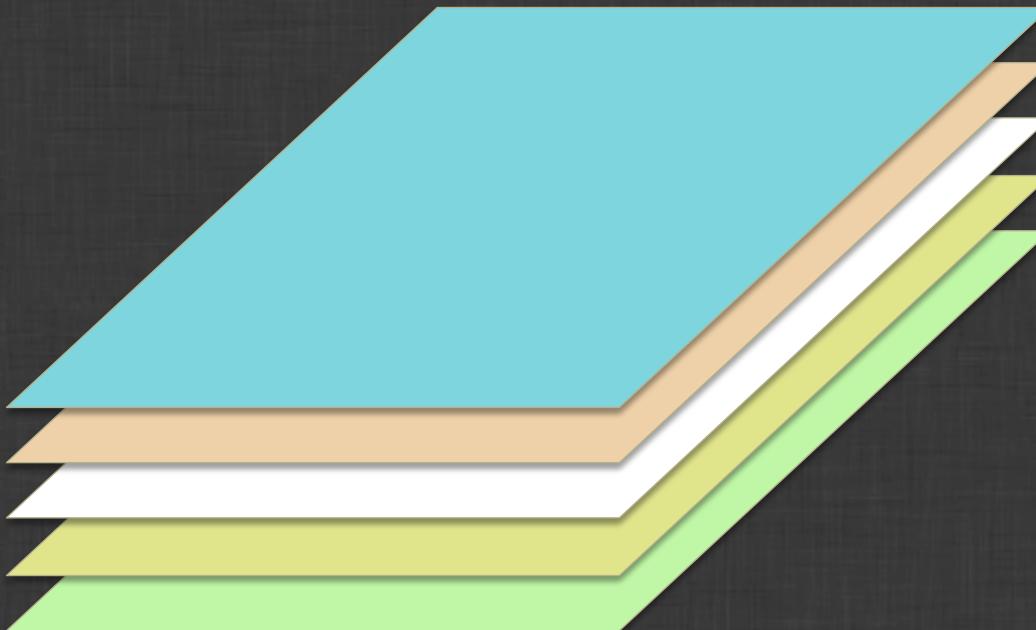
# For designers

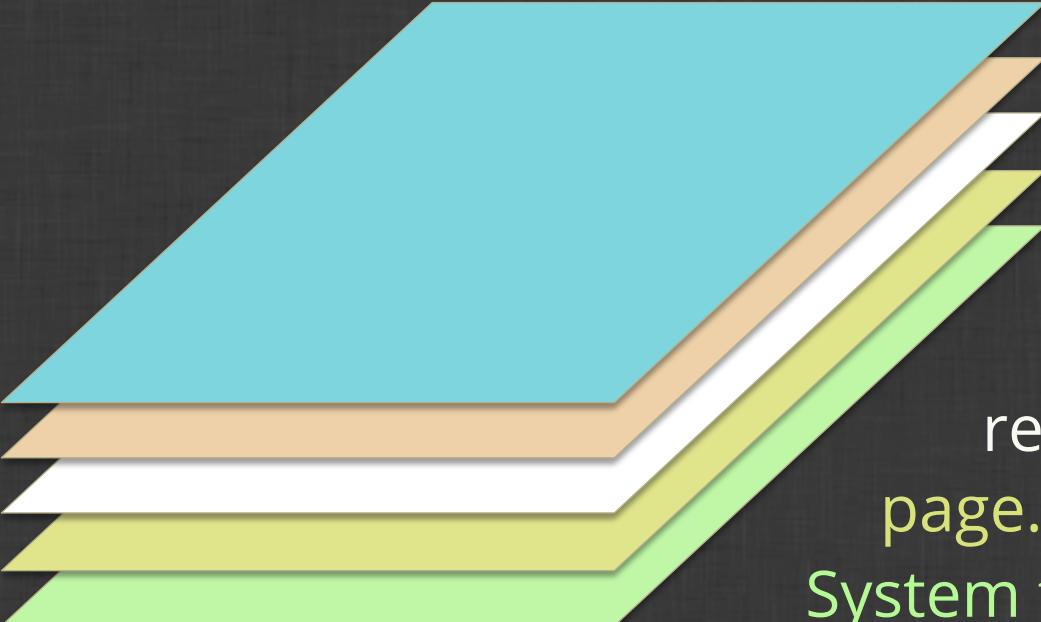
- But also: Envision designs and wireframes using a stacked-paper model to evaluate regions and the structure of the document.



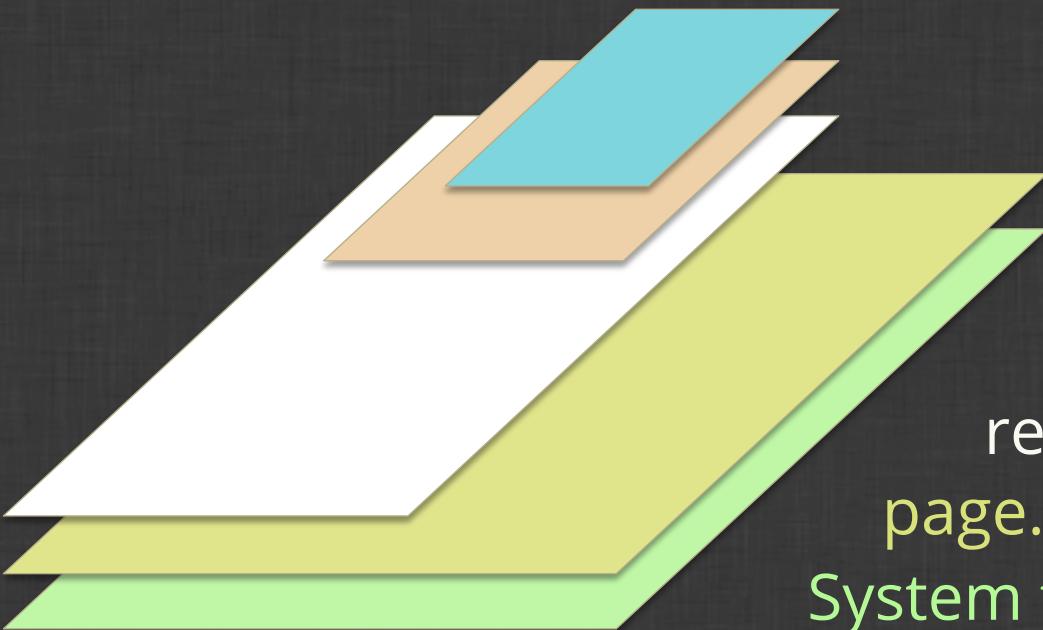
# For designers

- But also: Envision designs and wireframes using a stacked-paper model to evaluate regions and the structure of the document.

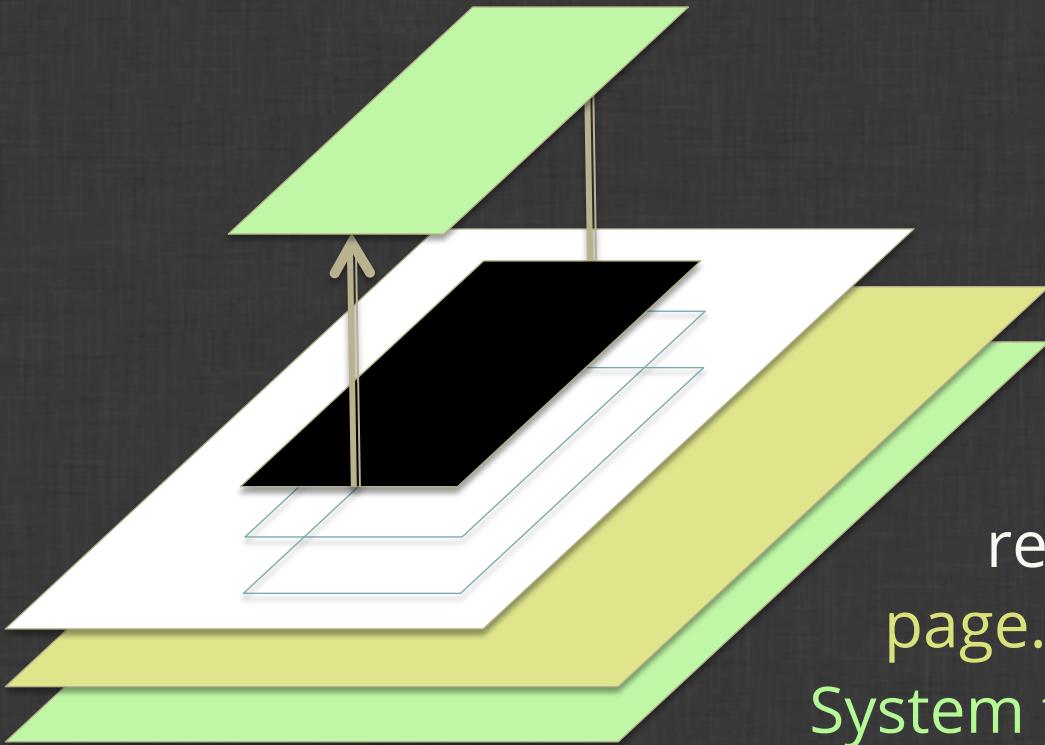




Block content  
block.tpl.php  
region.tpl.php  
page.tpl.php  
System template



Block content  
`block.tpl.php`  
`region.tpl.php`  
`page.tpl.php`  
System template



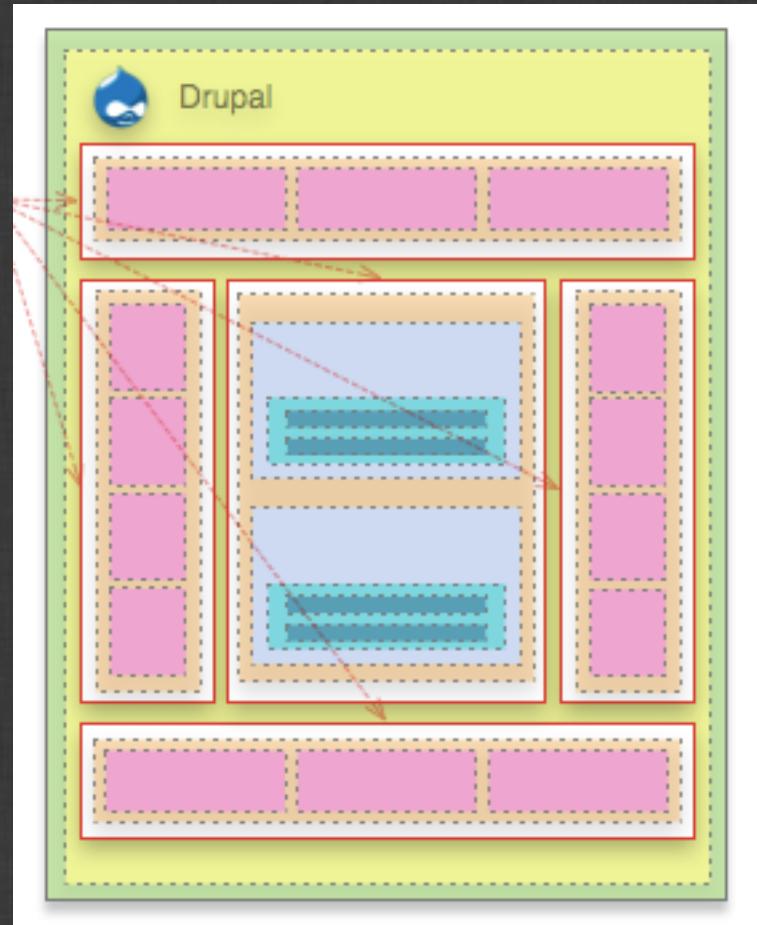
Block content  
**System** block.tpl.php  
region.tpl.php  
page.tpl.php  
System template



# For designers

If it ain't broke, don't fix it.

- Drupal gives us a very robust set of default system templates to work with.
- See what Drupal produces; it has impact on your design.



# 4

## Drupal-aware theming

Theme robustness

Conditions in theming

Best practices for templates

# For designers and themers

- We've all seen stuff like this before:

```
/* @group Drupal overrides */
.node-promoted h2,
#block-system-main-menu h2,
.node-promoted ul.links li {
  display: none;
}
/* @end */
```

# For designers and themers

- Drupal can sometimes produce some nasty overlaps in code with system CSS.
- Use a CSS reset or base presentational aspects of theme off of system defaults.
- Drupal overrides should generally be avoided, because they are hard to manage and uglify your beautifully written CSS.

# For designers and themers

- Use built-in Drupal display configurations to change the visibility and style of elements such as fields, labels, headings, etc.
  - For example, you can use Views to add CSS classes that are easier to manipulate than unwieldy classes like `.views-views-row` or `.views-views-field`.

# For designers and themers

- Use a style guide to manage hierarchical relationships between elements and to track how they are used on the page.
  - For example, track heading tags based on other headings in use as well as Drupal defaults.
  - Ken Woodworth, “Designing and Implementing Beautiful Flexible Interfaces,” Design 4 Drupal 2012.
  - [boston2012.design4drupal.org/sites/default/files/slides/Designing%20Interfaces.pdf](http://boston2012.design4drupal.org/sites/default/files/slides/Designing%20Interfaces.pdf)

# Theme robustness

- Key to Drupal is the split between *structure* and *presentation*.
- Keep as much as possible in the presentation layer so that more flexible structures can be accepted.

# For designers and themers

- Avoid esoteric code.
- Esoteric code leads to difficulties in markup handover and extending your prior work.
- As we design, whether it is a wireframe or prototype, we should think always about the next steps for the design, as if code will be reused every step of the way.

# For themers

- Take the following snippet from economist.com. How would you create this if the entire area was a region?

**Debate**

**Euro debt**

**Should the euro zone's debt be mutualised? »**

**Video**

Visit the library of weird materials, from aluminium foam to translucent concrete

**Watch »**

**The Economist**

INSIDE: A 14-PAGE SPECIAL REPORT ON THE GAS BOMBS  
China's next-best-one leader? | Good Lord! A crisis for Germany | How Merkel still dominates Germany | Basics and the art of innovation | Russia's industrial revolution

**Comeback kid**  
Rebuilding America's economy



**Full contents** | **Subscribe**

# Theme robustness with CSS

- When creating themes, we should avoid adding anything to the extreme amount of HTML Drupal already has. Two regions?

```
<div id="left">
  <div class="block"></div>
  <div class="block"></div>
</div>
<div id="right">
  <div class="block"></div>
</div>
```

# Theme robustness with CSS

- We should allow our HTML to embrace semantic richness as much as possible, as little as Drupal allows us:

```
<div class="block"></div>
<div class="block"></div>
<div class="block"></div>
```

# Theme robustness with CSS

- Instead, as much as possible, we should maintain semantic value in everything we write in template files.

```
.block {  
    float: left;  
}  
.block:last-child {  
    float: right;  
}
```

# Theme robustness with CSS

- We should also strive to avoid making modifications where they would impact usability or accessibility. *Design for the 80%.*

```
<div class="block">
  <h2 class="block-title">Title</h2>
  <div class="block-content">
    <p>A block.</p>
  </div>
</div>
```

# Theme robustness with CSS

- Let's reverse the order of our block elements.  
Easy enough in the template, right?
- Sometimes the easy way is not the right one.

```
<div class="block">
  <div class="block-content">
    <p>A block.</p>
  </div>
  <h2 class="block-title">Title</h2>
</div>
```

# Theme robustness with CSS

- If the modification *compromises the semantic structure or accessibility* of the document, then we should unequivocally avoid it.

```
<div class="block">
  <h2 class="block-title">Title</h2>
  <div class="block-content">
    <p>A block.</p>
  </div>
</div>
```

# Conditions in templates

- Remember back in the day?

```
<table>
  <tr>
    <td width="33%">Left sidebar</td>
    <td width="67%">Content area</td>
  </tr>
</table>
```

# Conditions in templates

- A client asks you to remove the sidebar.

```
<table>
  <tr>
    <td width="33%">Left sidebar</td>
    <td width="67%">Content area</td>
  </tr>
</table>
```

# Conditions in templates

- Uh-oh.

```
<table>
  <tr>
    <td width="67%">Content area</td>
  </tr>
</table>
```

# Conditions in templates

- We can condition based on sidebar existence:

```
<?php if ($page['sidebar_first']): ?>
  <div id="sidebar-first" class="sidebar">
    <?php print
  >> render($page['sidebar_first']); ?>
  </div>
<?php endif; ?>
```

# Conditions in templates

- With CSS we can specify based on body classes:

```
body.two-sidebars {  
    min-width: 980px;  
}  
  
body.sidebar-first,  
body.sidebar-second {  
    min-width: 780px;  
}
```

5

# To Drupal and beyond

Standards, conventions, documentation

Contributing to Drupal

New horizons

# Code standards

- Code standards in Drupal are important for code readability and documentation purposes.
- Writing a theme that you intend to contribute to Drupal requires awareness of the conventions involved.

# Code standards

- Indents consist of 2 spaces.
- Keep CSS readable and navigable (include line breaks).
- Alphabetize CSS properties within declaration blocks (a Drupal best practice).

# Code standards

- Refer to other themes, such as core themes and the most widely used contributed themes, to brush up on standards.

[drupal.org/coding-standards](http://drupal.org/coding-standards)

# Doxygen and documentation

- The Drupal implementation of Doxygen, a documentation aggregation service, has conventions that should be followed.
- Your themes should be well-commented and easy to navigate.

[drupal.org/node/1354](http://drupal.org/node/1354)

# Doxygen and documentation

- Comment blocks are as follows:

```
/**  
 * Documentation goes here.  
 */
```

# Doxygen and documentation

- Each file begins with a doc header as follows:

```
/**  
 * @file  
 * The theme system, which controls the  
 >> output of Drupal.  
 */  
...
```

# Doxygen and documentation

- And a listing of the variables in use:

```
...
* Available variables:
* - $forums: An array of forums to
>> display. Each $forum in $forums
>> contains:
*   - $forum->is_container: Is TRUE if
>> the forum can contain other forums.
...
```

# Contributing a theme

- Drupal is constantly in need of better, more flexible themes.
- Contributed themes, since they are optimized for Drupal, should follow a Drupal-only paradigm, but you can certainly account for other potential outcomes.

[drupal.org/node/14208](http://drupal.org/node/14208)

# The Spark project

- The upcoming Spark distribution will fundamentally change the way themes create layout, as well as the end-user UI.
- The possibility of flexible layouts is essential here. Themes will need to account for any possibility of layouts.

# The Spark project

- The goal of our Layout module is to give designers easier tools to create responsive layouts using grid systems.
- It will also give content editors much more capability in structuring their copy.

# The Spark project

- Please consider contributing to our efforts on Spark by testing and reporting issues.

[drupal.org/project/spark](http://drupal.org/project/spark)

[drupal.org/project/edit](http://drupal.org/project/edit)

[drupal.org/project/layout](http://drupal.org/project/layout)

- Kevin O'Leary, "Improving the Content Editing Experience with Spark", 10:30, Acquia Skyline
  - Demo: [www.youtube.com/watch?v=Ek2eyWZPl1c](https://www.youtube.com/watch?v=Ek2eyWZPl1c)

- Design principles can help guide us in creating themes that are flexible and extend well.
- Drupal-aware design involves consideration of the way Drupal structures its themable output.
- Drupal-aware theming involves consideration of structure vs. presentation and the reusability of theme code.

# Food for thought

- What do you think about accounting for when you write a theme?
- What methodologies do you use in your design and theming processes?
- What problems have you encountered during these processes?

# Thank you!

- **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](http://www.prestonsodesign.com)  
[drupal.org/user/325491](http://drupal.org/user/325491)  
pso@college.harvard.edu  
preston.so@acquia.com