



There's a Module For that!

Don't Use it!





Ted Bowman
a.k.a tedbow



- Entityform & Webform Conditional Modules
- Drupal Trainer
- DrupalEasy Podcast





- Drupal Training and Consulting
-  @sixmilettech
- sixmilettech.com





Contact Us

- Custom Drupal Training
- Project Mentoring
- Online Drupal Consulting





sixmiletotech.com

- Training Newsletter Signup
- Free Webinars
- Online Class discounts





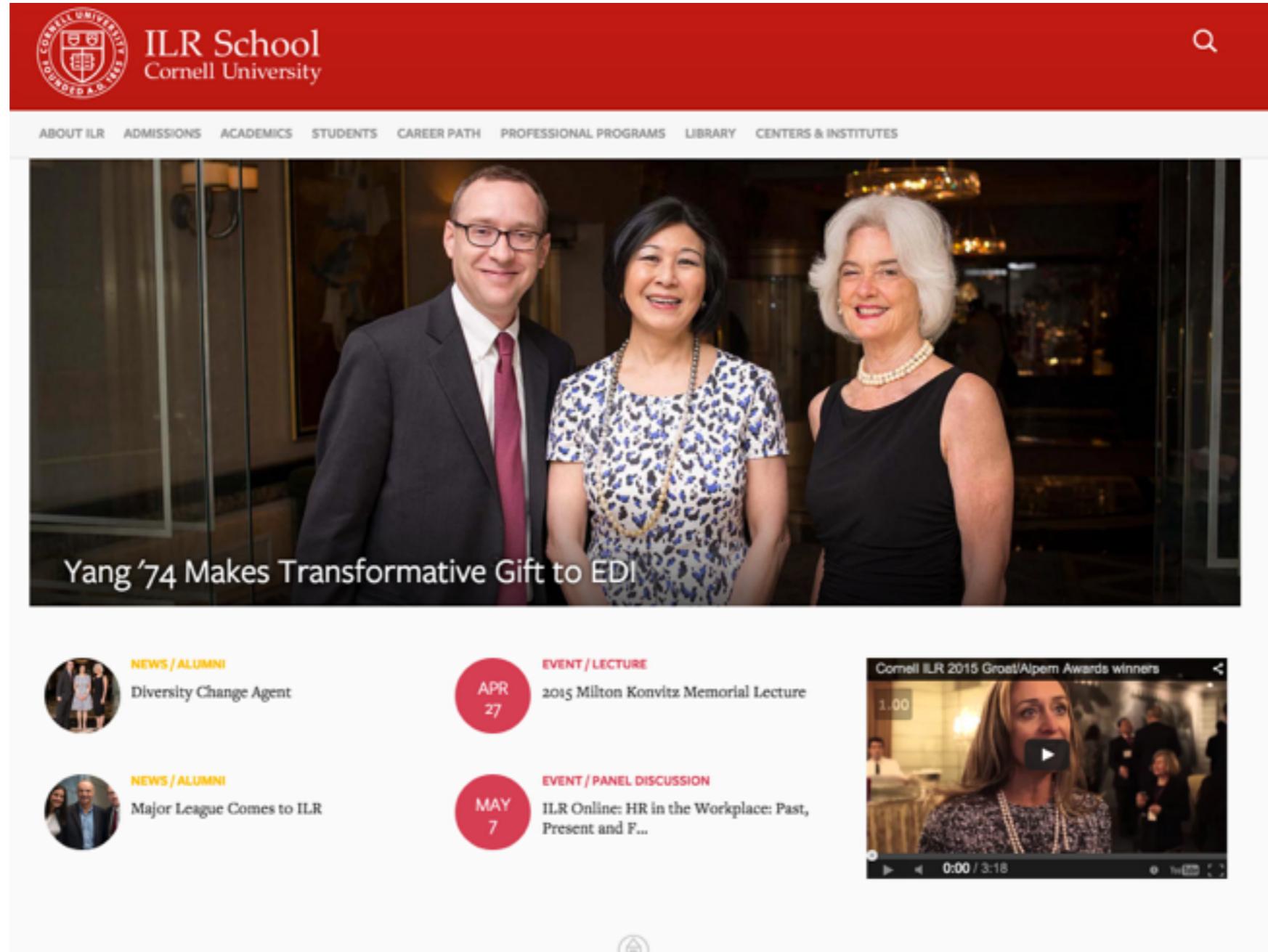
Drupal 7 Module Develop

- Online Professional Level Course
- Learn by doing
- sixmilettech.com/mod-dev





Cornell ILR



The screenshot shows the homepage of the ILR School Cornell University website. At the top, there is a red header bar with the Cornell University logo and the text "ILR School Cornell University". A search icon is located in the top right corner. Below the header, a navigation menu includes links for "ABOUT ILR", "ADMISSIONS", "ACADEMICS", "STUDENTS", "CAREER PATH", "PROFESSIONAL PROGRAMS", "LIBRARY", and "CENTERS & INSTITUTES". The main content area features a large photograph of three people (two men and one woman) smiling. Below the photo, the text reads "Yang '74 Makes Transformative Gift to EDI". On the left side, there are two news items: "Diversity Change Agent" (with a thumbnail image of two people) and "Major League Comes to ILR" (with a thumbnail image of a group of people). On the right side, there are two event listings: "2015 Milton Konvitz Memorial Lecture" (date: APR 27) and "ILR Online: HR in the Workplace: Past, Present and F..." (date: MAY 7). To the right of the events is a video player showing a woman speaking at an awards ceremony, with the text "Cornell ILR 2015 Groat/Alpem Awards winners". The bottom of the page features a small navigation bar with icons for home, search, and other site functions.

ILR School
Cornell University

ABOUT ILR ADMISSIONS ACADEMICS STUDENTS CAREER PATH PROFESSIONAL PROGRAMS LIBRARY CENTERS & INSTITUTES

Yang '74 Makes Transformative Gift to EDI

NEWS / ALUMNI

Diversity Change Agent

APR 27

EVENT / LECTURE

2015 Milton Konvitz Memorial Lecture

NEWS / ALUMNI

Major League Comes to ILR

MAY 7

EVENT / PANEL DISCUSSION

ILR Online: HR in the Workplace: Past, Present and F...

Cornell ILR 2015 Groat/Alpem Awards winners

1.00

0:00 / 3:18





Let's Talk

- Interested Drupal training or consulting?
- Meet up at the Camp
- ted@sixmilettech.com





There's a Module For that!

Don't Use it!





Disclaimer:

This presentation is not meant offend any of the following:

- Awesome Contrib Developers!
- People who code
- People who don't code
- People who say “There's a module for that!”
- People who say “No custom code”
- or People who give “Thumbs Up”





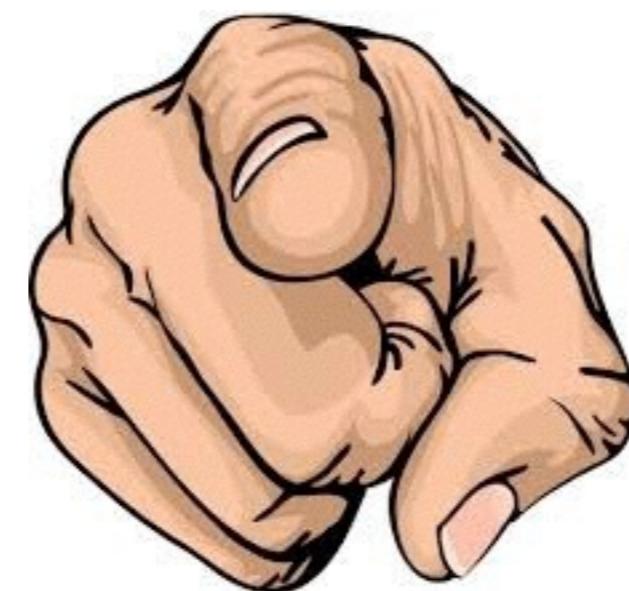
It is simply.

An attempt to start a discussion about why we do some of the things we do.





You





You have problem :(. . .

You would like

to show your nodes differently

. . . sometimes





Drupal has a solution!

▼ CUSTOM DISPLAY SETTINGS

Use custom display settings for the following views:

- Full content
- Teaser
- RSS
- Search index
- Search result highlighting input
- Tokens





Thanks Drupal!!!!





But you want more!!!

▼ CUSTOM DISPLAY SETTINGS

Use custom display settings for the following views:

Full content

Teaser

RSS

Search index

Search result highlighting input

Tokens

+ Featured





There's a module for that!

Entity view modes

View

Version control

Revisions

Automated Testing



We've introduced Drupal.org [Terms of Service](#) and [Privacy Policy](#). Please read both documents and [accept them on your user profile](#) to continue using Drupal.org.

Posted by [Dave Reid](#) on *August 2, 2011 at 10:56am*

The Drupal 7 successor to [Build modes](#) which will allow administrators to define custom view modes for entities. Custom entities are added to the entity registry via `hook_entity_info_alter()` so they are available to any code that uses `entity_get_info()` to provide a list of view modes for an entity. This includes node and user reference fields, Views, etc.

The screenshot shows the 'Entity View Modes' configuration page. It has two main sections: 'Content' and 'Node'. Each section contains a table with columns for 'VIEW MODE', 'TYPE', 'CUSTOMIZE', and 'OPERATIONS'. In the 'Content' section, there are two rows: 'Full content' (TYPE: Node, CUSTOMIZE: No, OPERATIONS: 'Edit content'), and 'Summary' (TYPE: Node, CUSTOMIZE: No, OPERATIONS: 'Edit content'). In the 'Node' section, there are five rows: 'Full content' (TYPE: Node, CUSTOMIZE: Yes, OPERATIONS: 'Edit content'), 'Raw' (TYPE: Node, CUSTOMIZE: Yes, OPERATIONS: 'Edit content'), 'Raw' (TYPE: Node, CUSTOMIZE: Yes, OPERATIONS: 'Edit content'), 'Summary' (TYPE: Node, CUSTOMIZE: Yes, OPERATIONS: 'Edit content'), and 'Summary' (TYPE: Node, CUSTOMIZE: Yes, OPERATIONS: 'Edit content').





Entity View Modes

Screenshot of the Drupal Configuration page for Entity View Modes.

The top navigation bar includes: Home, Dashboard, Content, Structure, Appearance, People, Modules, Configuration (selected), Reports, Help. The user is logged in as admin.

The breadcrumb trail shows: Home > Administration > Configuration > System > Entity view modes.

The Entity view modes interface displays two tables:

COMMENT

VIEW MODE	TYPE	CUSTOM SETTINGS	OPERATIONS
Full comment	In code	No	None (view mode locked)
Tokens	In code	No	None (view mode locked)

[Add new view mode](#)

NODE

VIEW MODE	TYPE	CUSTOM SETTINGS	OPERATIONS
Full content	In code	No	None (view mode locked)
Print	In code	No	None (view mode locked)
RSS	In code	No	None (view mode locked)
Search index	In code	No	None (view mode locked)
Search result	In code	No	None (view mode locked)
Teaser	In code	Yes	None (view mode locked)
Test bundle	In database	Yes	Edit Delete
Tokens	In code	No	None (view mode locked)

[Add new view mode](#)





Thanks Contrib!!!!





Let's make a View Mode!

The UI Makes it Easy

1. Configuration >> System >> Entity View
2. Add new view mode
3. Label = “Featured”
4. Export to Features
5. Local: git add >> git commit >> git push
6. Live: git pull
7. Done!





The UI makes it easy. Or does it?

```
 /**
 * Implements hook_entity_info_alter().
 */
function m_entity_view_mode_entity_info_alter(&$entity_info) {
  $entity_info['node']['view modes']['featured'] = array(
    'label' => 'Featured',
    'custom settings' => FALSE,
  );
}
```



- 
1. Configuration >> System >> Entity View
 2. Add new view mode
 3. Label = “Featured”
 4. Export to Features
 5. Local: git add >> git commit >> git push
 6. Live: git pull
 7. Done!

VS

1. 6 Lines of code
2. Local: git add >> git commit >> git push
3. Live: git pull
4. Done!





6 Lines of Code

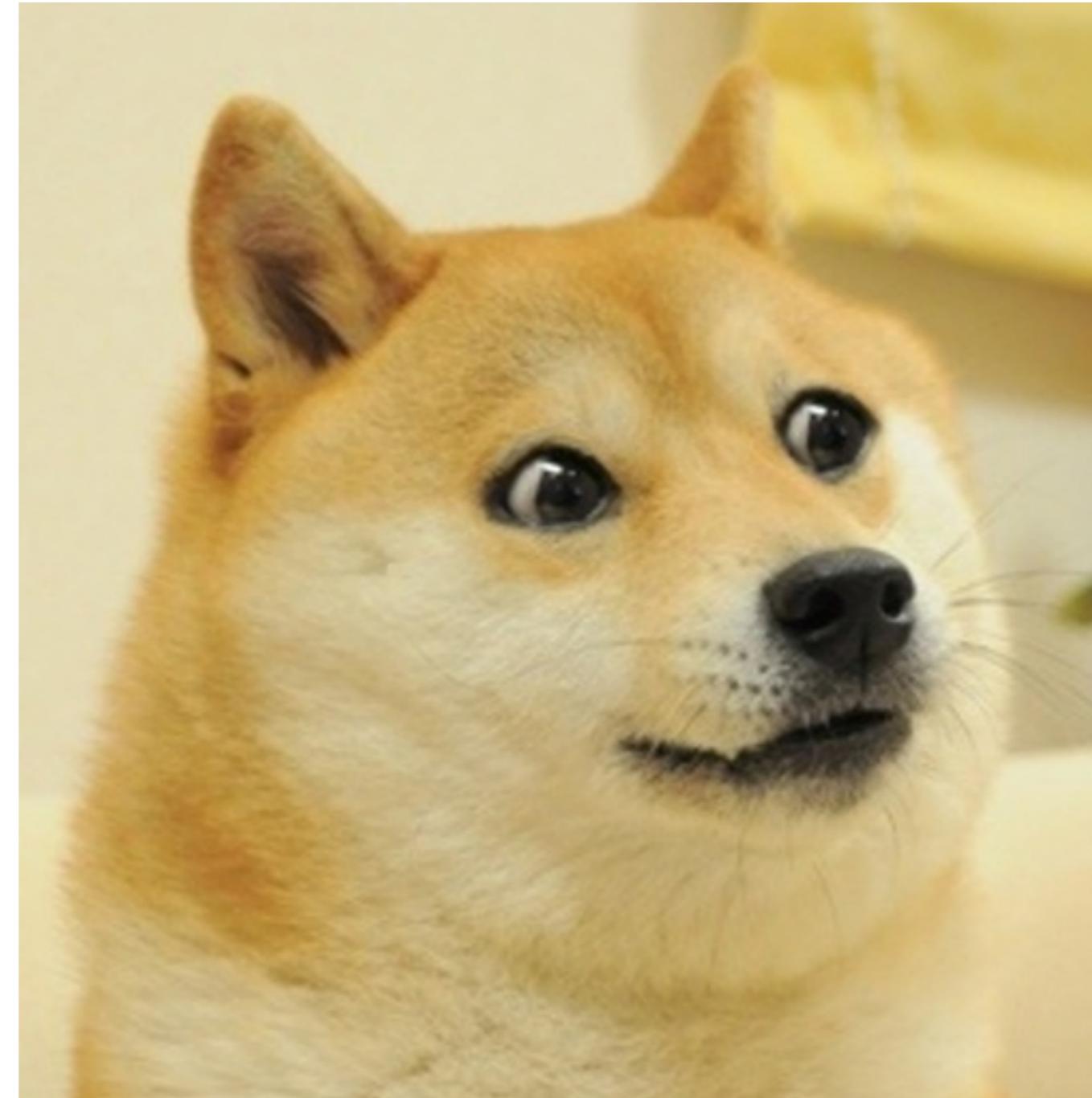
VS

~500 Lines of code
(Great Code)





Why so much Code?





Contrib Modules give you



Value





Flexibility Extra Features UI Interface



Extra Features

Do you need them?





Flexibility

Are you using it?





UI Interface

Do you need it?





At a cost. . . .
of
Complexity &
Performance



Did I mention bugs?

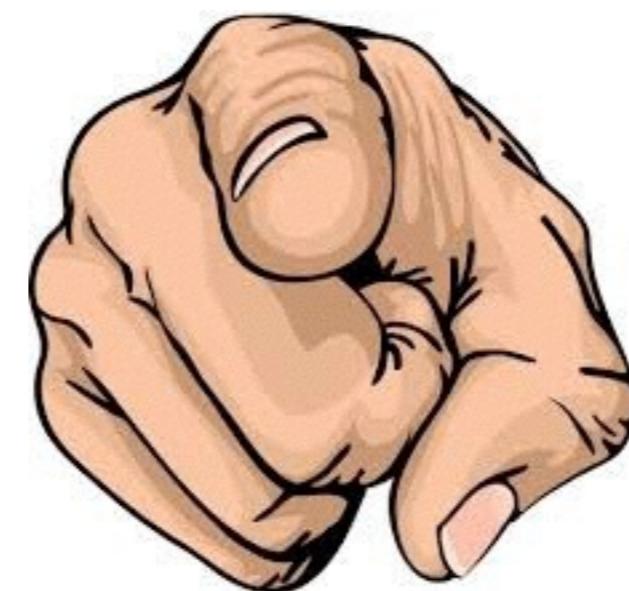




For Example



You





You have problem :(. . .

You would like

to override <title> tag

. . . per node





There a Module for that!

Page Title

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



We've [introduced](#) Drupal.org [Terms of Service](#) and [Privacy Policy](#). Please read both documents and [accept them on your user profile](#) to continue using Drupal.org.

Posted by [nicholasThompson](#) on February 21, 2006 at 8:46am

The word "title" is a bit overloaded. Every piece of content in Drupal has a title, and so does every page. The *page title* is the one found in the HTML head inside the <title> tag. It is also used on SERPs (Search Engine Result Pages) and can greatly enhance your websites SEO (Search Engine Optimization).

This module gives you granular control over the page title. You can specify patterns for how the title should be structured and, on content creation pages, specify the page title separately to the content's title.



Page Title





Thanks Contrib!!!!





But wait...
What about the cost?





The screenshot shows a code editor interface with the following details:

- Title Bar:** Shows the file path "page_title.module" and various tool icons.
- Left Sidebar:** Includes sections for "Project", "Structure", "Favorites", and "Documentation".
- Code Area:** Displays the following PHP code from the `page_title.module` file:

```
page_title_menu():array
26     $items = array(
27         t('<strong>Default Page</strong>: <samp><em>page title</em> | <em>site name</em></samp>'),
28         t('<strong>Default Frontpage</strong>: <samp><em>site name</em> | <em>site slogan</em></samp>'),
29     );
30     $output .= theme('item_list', $items, NULL, 'ol');
31     $output .= '<p>' . t('The <strong>Page Title</strong> module lets you change these defaults in two ways. First, you can adjust the pattern in the <em>site name</em> token. Second, you can use the <em>[current:page-title]</em> token which will default to the value returned from <samp>drupal_get_title</samp>().');
32     $output .= '<p>' . t('Certain types of page title pattern have access to special tokens which others do not, depending on their <em>scope</em>.');
33     $output .= '<p>' . t('break;');
34 }
35     return $output;
36 }
37 }

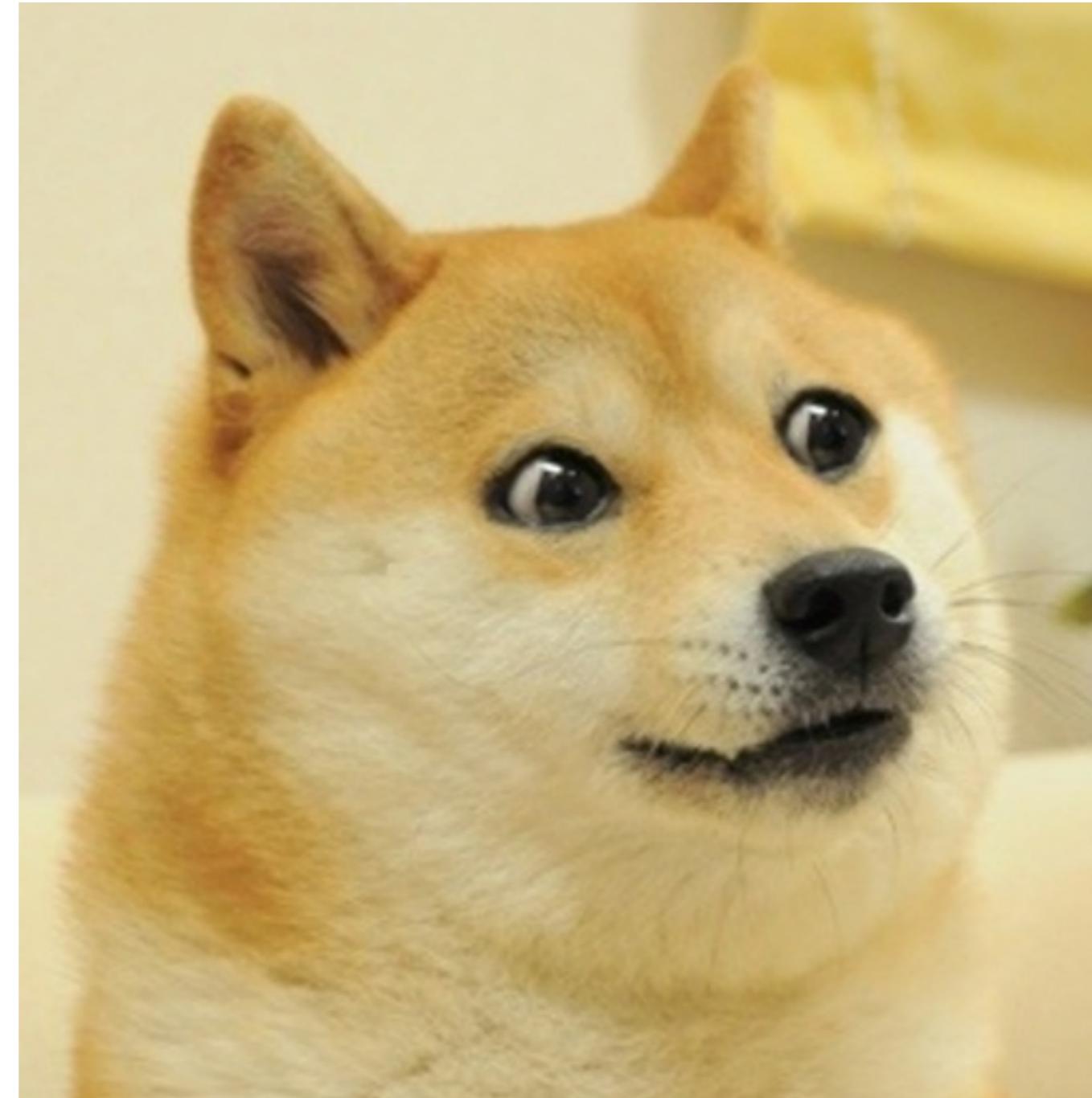
40 /**
41 * Implements hook_requirements().
42 */
43 function page_title_requirements($phase) {
44     $requirements = array();
45     if ($phase == 'runtime') {
46         // Are we on an old version?
47         if (!page_title_is_up_to_date()) {
48             $requirements['page_title_version'] = array(
49                 'title' => t('Page title version'),
50                 'value' => t('Out of date'),
51                 'description' => t('The Page Title module must be updated. You should run the !link immediately.', array('!link' => l(t('database update'), 'admin/settings/page-title/upgrade'))),
52                 'severity' => REQUIREMENT_ERROR,
53             );
54         }
55         // Nope - we're on the latest version
56     } else {
57         // Does the old table exist (it is left after the upgrade in case an admin wants to check the upgrade went ok)
58         if (db_table_exists('page_title_old')) {
59             $requirements['upgrade_table'] = array(
60                 'title' => t('Page Title upgrade table present'),
61                 'value' => '',
62                 'description' => t('The Page Title upgrade table (<code>page_title_old</code>) is present. You can remove it !link', array(
63                     '!link' => l(t('using this script'), 'admin/settings/page-title/drop-old-table'),
64                 )),
65             );
66         }
67     }
68 }
```

The code implements the `hook_menu()` and `hook_requirements()` hooks for the Page Title module.





Why so much Code?





Flexibility Extra Features UI Interface



Page Title implements Hooks(lots of hooks)

- hook_form_alter
- hook_node_load
- hook_node_insert
- hook_node_update
- hook_node_delete
- hook_taxonomy_term_update
- hook_taxonomy_term_delete
- hook_taxonomy_term_insert
- hook_user_insert
- and many more.....





At a cost. . . .
of
Complexity &
Performance



Did I mention bugs?





Page Title Features:

- Token Integration
- Granular Control
- Token Patterns per-content type
- Pagination Suffix
- Views Integration
- Terms Support





But what was your
problem?



You would like
to override <title> tag
...per node





```
/**  
 * Implements hook preprocess_html().  
 */  
function m_page_title preprocess_html(&$variables) {  
  if ($node = menu_get_object()) {  
    if (!empty($node->field_page_title[LANGUAGE_NONE][0]['value'])) {  
      $variables['head_title'] = $node->field_page_title[LANGUAGE_NONE][0]['value'];  
      $variables['head_title'] .= ' | ' . variable_get('site_name', '');  
    }  
  }  
}
```





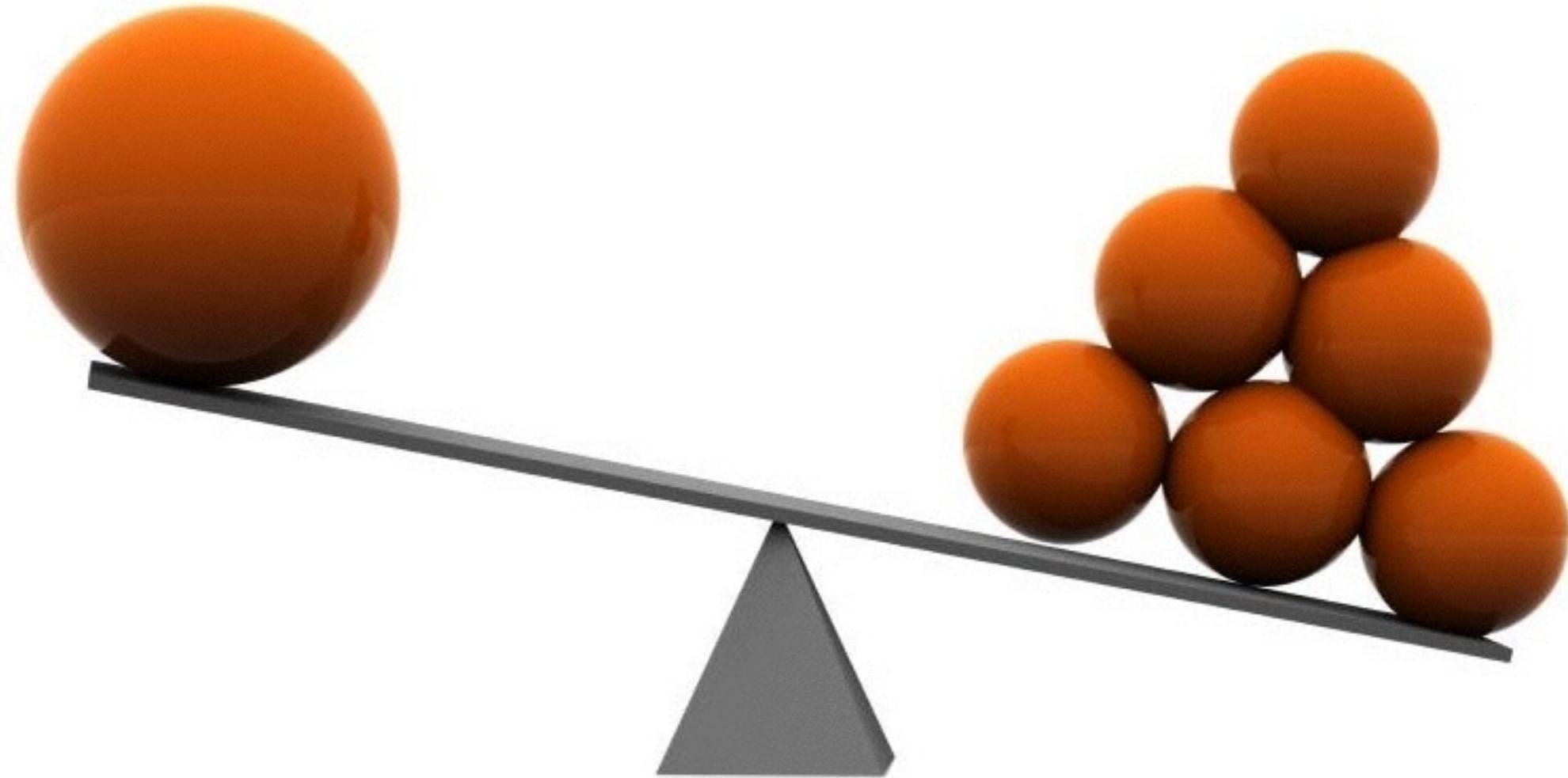
All the code you run

- + Performance Hit
- + Potential Bugs



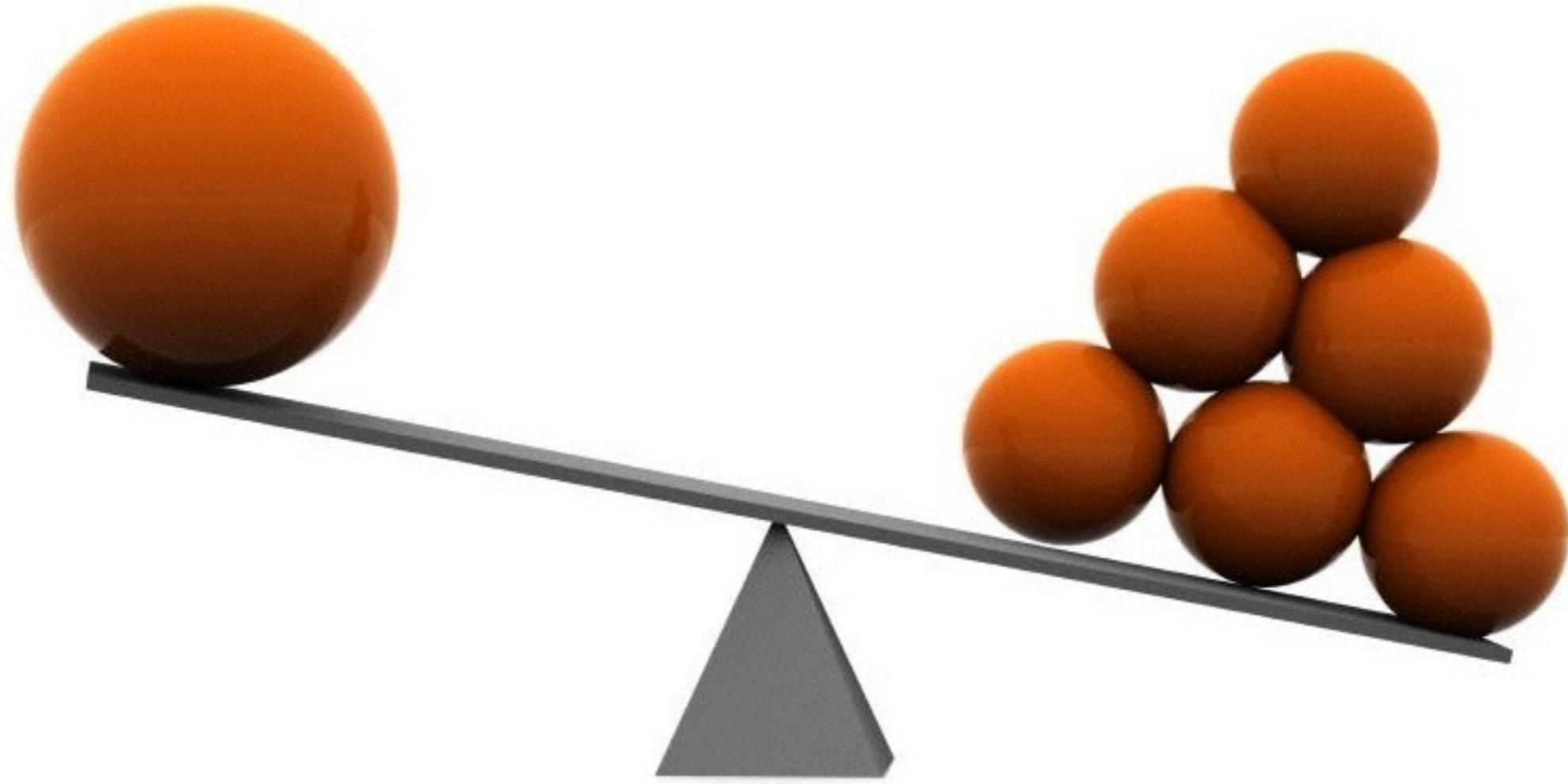


Balancing Act





Contrib Modules



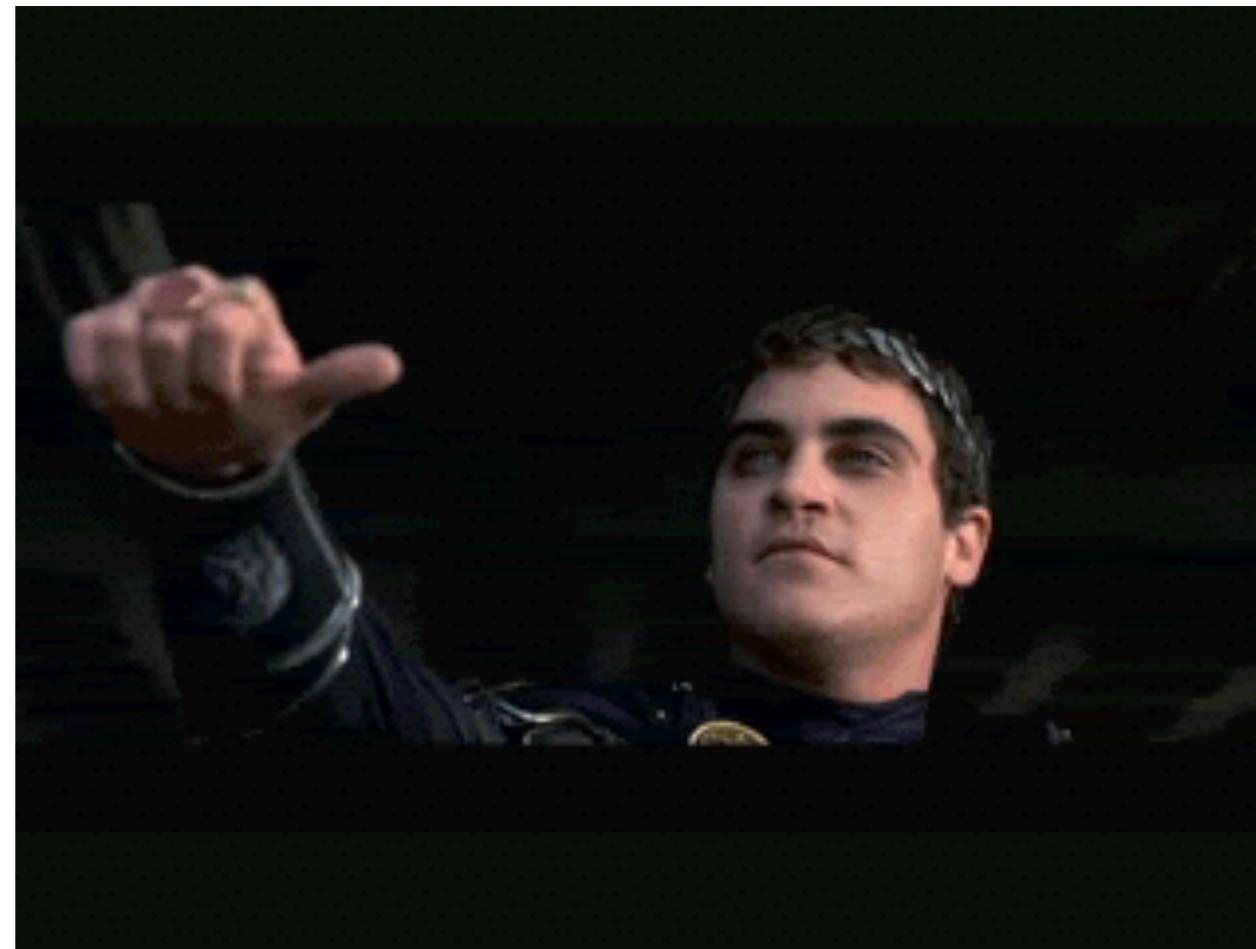
Custom Code





Custom Code

The solution to all our problems





WARNING

**MONSTERS
COULD OCCUR AT
ANY TIME**

©2008 SUNSTAR INDUSTRIES, INC.



@sixmilettech



Custom Code Has Costs

- You own it
- You maintain it
- You have to write it
- You have to upgrade it





Custom Code has benefits

- It's easy! (for some stuff)
- It's stable (for some stuff)
- It simpler (or can be)
- You understand it!
- It's fun!





Most Sites (of mine)

- Have at least 1 custom module
- `hook_form_FORM_ID_alter` most common hook
- For small things
- Avoid uninstalling complicated modules
- Save time





For Example

```
 /**
 * Implements hook_form_FORM_ID_alter().
 */
function MYMODULE_form_article_node_form_alter(&$fo
  global $user;
  if (user_access('edit any article content')){
    $form['field_location']['#access']= FALSE;
  }
}
```





Gets me:

- Access to field edit access based on existing permission
- Based on own permission + 5 lines





Avoids:

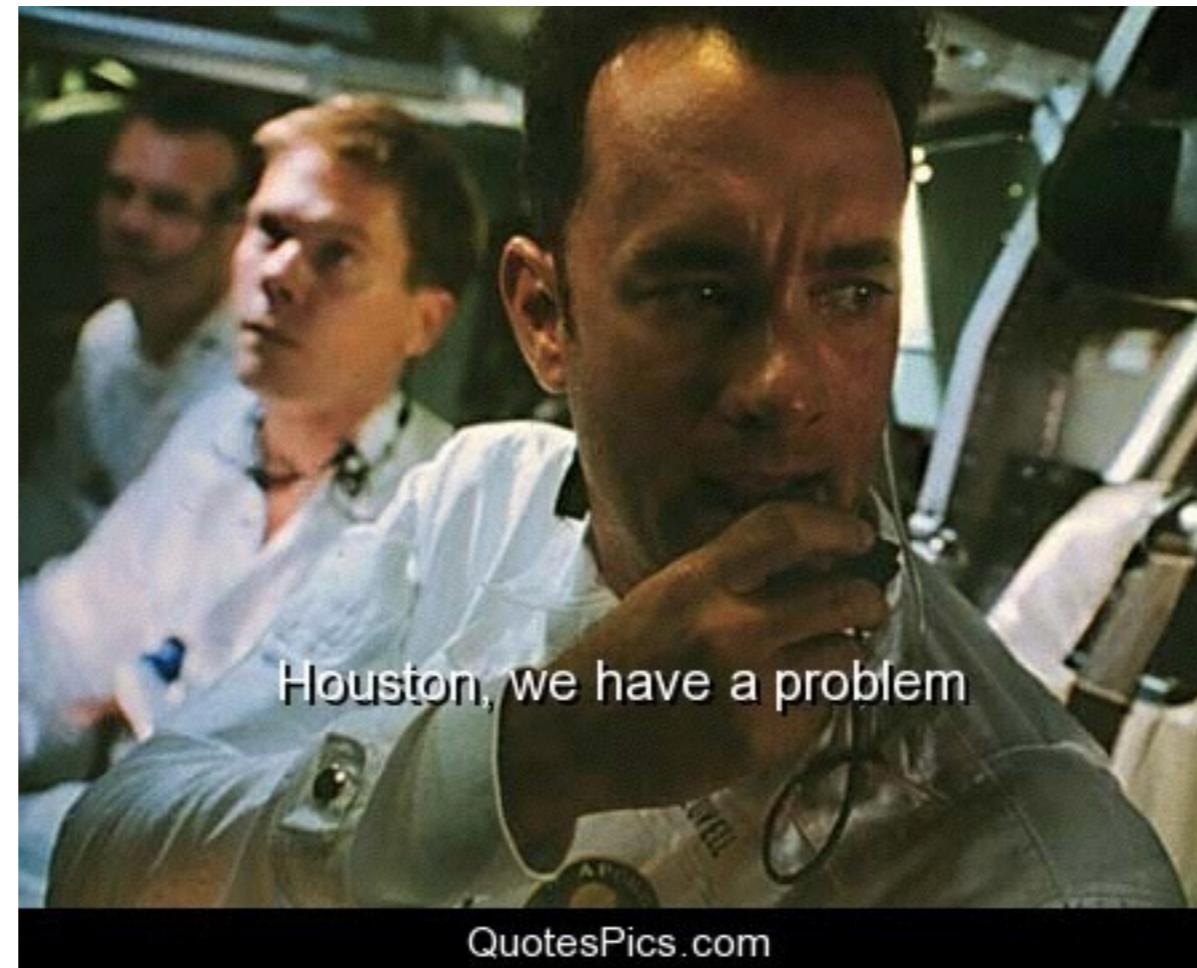
- Field Permission Module
- ~600 Lines code
- No per bundle permissions







But, we have problem





It's Hard Coded!!!

- Not Re-usable
- New code every project
- Old sites don't get improvements





HARD-CODED!

```
 /**
 * Implements hook_form_FORM_ID_alter().
 */
function MYMODULE_form_article_node_form_alter(&$fo
  global $user;
  if (user_access('edit any article content')){
    $form['field_location']['#access']= FALSE;
  }
}
```





HARD-CODED!

```
 /**
 * Implements hook_entity_info_alter().
 */
function m_entity_view_mode_entity_info_alter(&$entity_info) {
  $entity_info['node']['view modes']['featured'] = array(
    'label' => 'Featured',
    'custom settings' => FALSE,
  );
}
```





But, what if it wasn't

- Wasn't Hard-coded
- You didn't own it
- Configurable
- Re-usable





Introducing.....

Minimalist Drupal Modules

(they are small)





Minimalist Field Permissions

```
/**  
 * Implements hook_form_FORM_ID_alter().  
 */  
function m_field_permissions_form_node_form_alter(&$form, &$form_state, $fc  
$settings = variable_get('m_field_permissions_settings', array());  
if (isset($settings[$form['#bundle']])) {  
    foreach ($settings[$form['#bundle']] as $field_name => $permission) {  
        if (!user_access($permission)) {  
            $form[$field_name]['#access'] = FALSE;  
        }  
    }  
}  
}
```





Minimalist Entity View Modes

```
function m_entity_view_modes_entity_info_alter(&$entity_info) {
  $settings = variable_get('m_entity_view_modes', array());
  foreach ($settings as $entity_type => $view_modes) {
    if (isset($entity_info[$entity_type])) {
      $entity_info[$entity_type]['view modes'] += $view_modes;
    }
  }
}
```





Configurable???



But there's no UI!!!!





Configurable != UI





Where's the Config?





settings.php

```
/*
 * Conf file routing
 */
if (file_exists(dirname(__FILE__) . '/site_conf.php')) {
  include dirname(__FILE__) . '/site_conf.php';
}
```





site_conf.php

```
$conf['m_field_permissions_settings'] = array(  
  'article' => array(  
    'field_location' => 'edit own article content',  
  ),  
) ;
```

Minimalist Field Permissions





site_conf.php

```
$conf['m_entity_view_modes'] = array(  
  'node' => array(  
    'featured' => array(  
      'label' => 'Featured',  
      'custom settings' => FALSE,  
    ),  
    ),  
  'user' => array(  
    'monthly' => array(  
      'label' => 'User of Month',  
      'custom settings' => FALSE,  
    ),  
    ),  
);
```

Minimalist Entity View
Modes





Add another module

One Config File

Add To Git

No Need for Features

(You will still use Features)





Minimalist Gmap

FIELD	LABEL	FORMAT
⊕ Location	Above	✓ Embedded Google Map Default Plain text Trimmed <Hidden>
⊕ Image	<Hidden>	
⊕ Body	<Hidden>	Default
⊕ Tags	Above	Link





```
function m_gmap_field_formatter_info() {
  return array(
    'm_gmap_simple' => array(
      'label' => t('Embedded Google Map'),
      'field types' => array('text', 'text_long', 'text_with_summary'),
    ),
  );
}

/**
 * Implements hook_theme().
 */
function m_gmap_theme() {
  return array(
    'm_gmap_simple' => array(
      'variables' => array('text' => '', 'api_key' => NULL),
      'template' => 'm-gmap-simple'
    ),
  );
}

/**
 * Implements hook_field_formatter_view().
 */
function m_gmap_field_formatter_view($entity_type, $entity, $field, $instance, $langcode, &$items,
  $settings = $display['settings'];
  $element = array();
  switch ($display['type']) {
    case 'm_gmap_simple':
      foreach ($items as $delta => $item) {
        $element[$delta] = array(
          '#theme' => 'm_gmap_simple',
          '#text' => urlencode($item['value']),
          '#api_key' => variable_get('m_gmap_api_key', NULL),
        );
      }
      break;
  }
  return $element;
}
```





site_conf.php

```
$conf['m_gmap_api_key'] = 'AIzaSyCWthzaH2ZtTGMidl3KHfTVrxBuoyCuxYo';
```

Minimalist Field Permissions





site_conf.php

- Code but re-useable
- Still simple
- Re-use within Organization





Re-use as Contrib Module?

- It's not the “Drupal Way”?
- Is that a good reason?





But I am not a coder?

```
$conf['m_entity_view_modes'] = array(  
  'node' => array(  
    'featured' => array(  
      'label' => 'Featured',  
      'custom settings' => FALSE,  
    ),  
    ),  
  'user' => array(  
    'monthly' => array(  
      'label' => 'User of Month',  
      'custom settings' => FALSE,  
    ),  
    ),  
);
```

Do you need to be a Coder?





DRUPAL TRAINING AND DEVELOPMENT

Questions?





DRUPAL TRAINING AND DEVELOPMENT

Contact Us

- Custom Drupal Training
- Project Mentoring
- Online Drupal Consulting

sixmilettech.com

