

Taller de Drupal – Sesión 1

Instalación de Drupal

Descarga de drupal.org

Instalar utilizando el repositorio **CVS**

Instaladores de tipo “**stack**” Bitnami / Acquia

Fantastico

Instalación de Drupal

Copiar el fichero default.settings.php en settings.php (sites/default)

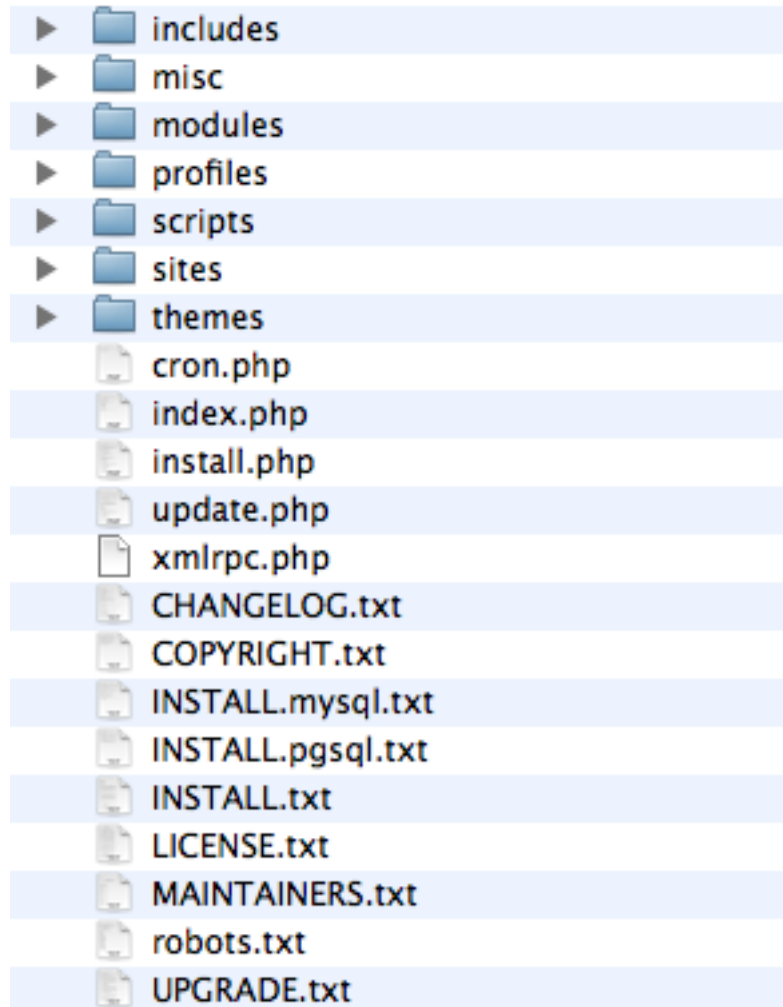
Modificar los permisos del fichero settings.php para que sea modificable por todo el mundo.

Crear la base de datos y asignarle permisos al usuario.

Seguir el proceso de instalación.

Restaurar los permisos del fichero settings.php.

Estructura de directorios



Administración de Drupal Contenido

- El contenido se organiza en unidades de información llamadas **nodos**.
- A través de los **tipos de contenido**, se pueden asignar diferentes comportamientos a los distintos tipos de nodos.
- El contenido se categoriza a través de las **taxonomías**.

Administración de Drupal Usuarios

- La gestión de usuarios en Drupal está **integrada** en el interfaz de administración.
- Los usuarios se agrupan en **roles**, que definen los límites que tiene cada grupo de usuario.
- La **granularidad** de permisos se asigna a través de los roles.

Administración de Drupal

Construcción del sitio

- Instalación y habilitación de **módulos**.
- Instalación de **plantillas** o temas.
- Control de **menús**.
- Gestión de los **bloques** de información y las regiones en los que se muestran.
- Configuración de los **alias de URL**.

Administración de Drupal

Configuración del sitio

- Normalmente los módulos, tanto del core como contribuidos, exponen sus opciones de configuración en este menú.
- Opciones de guardado de **ficheros**.
- Opciones de **rendimiento** y caché.
- Información del sitio
- Opciones de **formato de entrada**.
- Configuración de fecha.

Administración de Drupal Informes

- El **informe de estado** muestra si hay algún problema con nuestro sitio.
- Muchos módulos implementan entradas de log para informar de errores o eventos importantes: **watchdog**.
- Consulta de **actualizaciones** disponibles, para el core o para módulos contribuidos: update status.

Taller de Drupal – Sesión 2

Extendiendo Drupal

Módulos contribuidos

- Los módulos contribuidos extienden las funcionalidades de Drupal sin que sea necesario modificar el código del sistema.
- Es posible programar módulos propios que se ajusten a las necesidades de la web.
- También existen centenares de módulos de terceros contribuidos bajo licencia GPL 2 que pueden utilizarse y adaptarse libremente.

Módulos contribuidos ¿Dónde conseguirlos?

- La comunidad de Drupal está centralizada en drupal.org, así como los módulos y las plantillas.
- No existen en la actualidad módulos de pago.
- Otros recursos, como drupalmodules.com ayudan a localizar los módulos contribuidos que se adapten a nuestras necesidades.

Módulos contribuidos

¿Cómo elegirlos?

En ocasiones, podemos encontrar varios módulos que se solapan en funcionalidad ¿cómo elegirlos?

- **Estadísticas de uso:**
<http://drupal.org/project/usage/views>
- **Cola de issues:**
<http://drupal.org/project/issues/views?status=All&categories=All>
- **Similar Module Review:**
<http://groups.drupal.org/similar-module-review>

Módulos contribuidos

Instalación

- Descargar de la página del módulo en drupal.org.
- Descomprimir
- Copiar en el directorio de nuestra aplicación, por ejemplo `sites/all/modules`

Módulos contribuidos

Habilitación

- Acceder con un usuario con permisos a Construcción del sitio > Modulos
- Seleccionar el módulo a habilitar.
- Guardar.

Principales Módulos contribuidos: **CCK**

- **CCK** = Content Construction Kit
<http://drupal.org/project/cck>
- Permite **crear nuevos tipos de contenido** con los campos que necesitemos **desde el interfaz.**
- También es posible **extender los tipos de contenido** que otros módulos instalan o los que vienen por defecto.
- Provee una API para que **otros módulos creen tipos de campos.**

Extensiones de CCK

Hay muchos otros módulos contribuidos que extienden CCK y nos permiten crear campos con determinados tipos de datos.

- Date
- Location
- Imagefield / Filefield
- Link
- Calendar

Principales Módulos contribuidos: **Views**

- Proporciona una **interfaz flexible** para presentar listados de información a los usuarios.
- Permite a los administradores de sitios **Drupal generar listados** de contenido, usuarios, galerías de imágenes, búsquedas personalizadas y mucho más **sin tocar código**.
- Es el módulo contribuido **más utilizado** para proyectos **Drupal** (más de 60,000 sitios lo utilizan)

Usos habituales de Views

- Todo tipo de listados estáticos o dinámicos de nodos y de usuarios ordenados y filtrables.
- Galerías de imágenes, carruseles, slideshows, etc.
- Búsquedas personalizables.
- Exportación de contenido en formato RSS, XML, RDF, etc.

Razones para usar Views

- Facilidad para generar listados sin escribir código. Efecto “caja negra”.
- Reutilización de vistas completa o parcialmente.
- Integración con CCK, Panels y otros módulos.
- Filtros expuestos
- Diferentes tipos de vistas para cambiar la presentación de los datos.
- Documentación y soporte

Razones para NO usar Views

- Control absoluto sobre los datos recuperados y el código ejecutado.
- Optimización de las consultas.
- Control de versiones, auditoría, ...
- ~~Dificultad de manejo.~~

Extensiones de Views

Gracias a su completo API, el módulo Views dispone de varias extensiones que complementan su funcionamiento tanto a nivel gráfico como a nivel de los datos que se muestran.

Views Bonus, Views Tabs, Views Slideshow,
Views Accordion, Views attach,
Views Datasource, Views Popup

Principales Módulos contribuidos: **ImageCache**

- Se utiliza para cachear las imágenes y mostrarlas modificadas en diferentes tamaños, escalas, etc partiendo de un único original.
- Requiere el módulo Image API y una librería de manejo de imágenes, GD o Imagemagik.
- A través de otros módulos, como ImageCache Effects o ImageCache Actions es posible poner marcas de agua, reflejos...

Principales Módulos contribuidos: **Pathauto y Token**

- Pathauto se utiliza para generar urls semánticas de forma automatizada, basándose en cadenas de substitución.
- Token proporciona estas cadenas de substitución, para realizar los reemplazos, utiliza variables de sistema o también otras proporcionadas por módulos contribuidos.

Principales Módulos contribuidos: Panels

- Hace posible **crear estructuras de páginas** desde el interfaz de administración.
- Es posible **incluir** casi **cualquier elemento dentro** de los paneles, vistas, nodos, cck, bloques...
- Facilita el **diseño de páginas** con estructuras complejas.

Principales Módulos contribuidos: **Control de flujo**

Existen varios módulos para realizar tareas basadas en eventos, algunos de ellos, como **Triggers** y **Actions** están incluidos en el núcleo, pero se puede extender su funcionalidad utilizando los módulos **Rules** o **Workflow** para eventos y acciones avanzadas.

Principales Módulos contribuidos: **SEO**

Drupal tiene soporte en el núcleo para ciertas **mejoras de posicionamiento**, como las URL limpias, pero hay módulos contribuidos que ayudan a esta funcionalidad, entre otros:

- **Xmlesitemap** genera un mapa del sitio en XML.
- **MetaTags** proporciona tags a cada nodo.
- **Path Redirect** ayuda a realizar redirecciones a través del interfaz.
- **Search404** proporciona páginas de búsqueda para los errores 404.

Principales Módulos contribuidos **WYSIWYG**

- Fck Editor
- TinyMCE
- BUEditor
- YUI Editor
- El futuro: WYSIWYG API

Otros módulos contribuidos

- [Devel](#) – ayuda al desarrollo
- [Internacionalización](#) – Sitios multilingües
- [Flag](#) – Marcar contenido
- [Multimedia](#)
 - [Embedded Media Field](#) – Campo CCK
 - [Audio](#)
 - [Video](#)

Taller de Drupal – Sesión 3

Extendiendo Drupal

Desarrollo de módulos

Un módulo es un trozo de código que se integra con el resto de una aplicación en Drupal a través del API del núcleo o de otros módulos.

Es posible desarrollarlos desde cero o partir de otros módulos ya desarrollados y modificar éstos o utilizarlos como base.

Estructura de ficheros de un módulo

demo.info

demo.module

demo.install

Normalmente se utilizan también los ficheros README.txt y CHANGELOG.txt para documentación, además de ficheros .inc para inclusiones de otro código, css, js...

Fichero .info

El fichero .info define los metadatos del módulo:

- Nombre
- Descripción
- Versión de Drupal
- Dependencias
- Paquete al que pertenece

Debe estar en el mismo directorio y tener el mismo nombre que el el fichero .module

Fichero .install

En este fichero opcional se codifican las operaciones necesarias para el funcionamiento inicial del módulo, normalmente la creación del modelo de datos, variables, etc.

Debe incluir dos hooks: `hook_install` y `hook_uninstall` que serán llamados en la instalación y desinstalación respectivamente.

Hooks



foto

Hooks

- Los hooks o “ganchos” indican a otros módulos de Drupal si nuestro módulo personalizado tiene algo que añadir y de esta forma se puede alterar el funcionamiento de una acción determinada.
- Se expresan como hook_* donde hook se reemplaza por el nombre del módulo.
- Los módulos invocan hooks utilizando la función `module_invoke_all('nombre_hook')`

Hooks principales: hook_menu()

- Define tanto elementos del menú como rutas que capturan las URLs llamadas (callback).
- Permite asignarle a determinadas rutas llamadas a funciones del módulo.
- Proporciona una capa de seguridad filtrando qué usuarios pueden acceder a las rutas.

Hooks principales: hook_perm()

- Define los permisos de los usuarios.
- Permite definir opciones de acceso que luego serán visibles en la página de gestión de permisos de los usuarios.

Hooks principales: hook_block()

- Declara un bloque o una serie de bloques que posteriormente pueden ser utilizados en el menú de administración correspondiente, de forma que se pueden asignar a regiones.
- Es posible configurar multitud de opciones, el peso, la región por defecto, las opciones de visualización, caché, etc.

Otros Hooks

- `hook_install / hook_uninstall`
- `hook_cron`
- `hook_user`
- `hook_nodeapi`
- `hook_theme`
- ...

Form API

- Permite mostrar formularios y realizar la validación y el envío de los mismos.
- Es posible modificar formularios ya existentes utilizando el hook `hook_form_alter`.
- Guía de Forms API:
http://api.drupal.org/api/file/developer/topics/forms_api.html/6
- Referencia de Forms API:
http://api.drupal.org/api/file/developer/topics/forms_api_reference.html

Taller de Drupal – Sesión 4

Drupal Theming

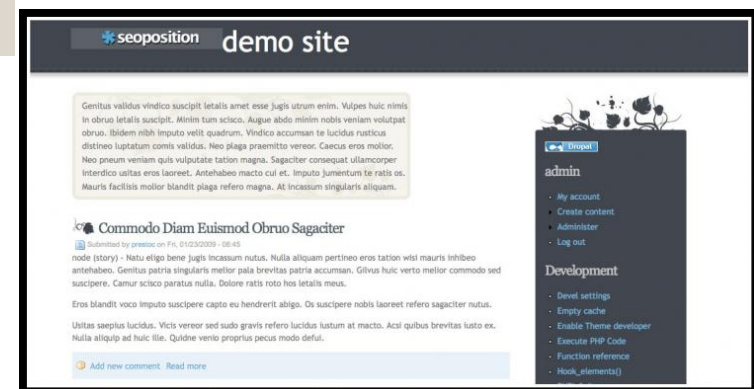
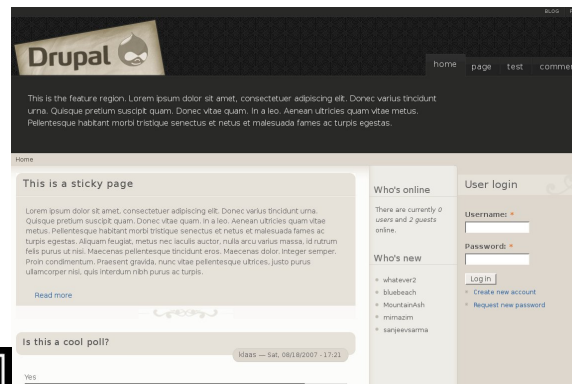
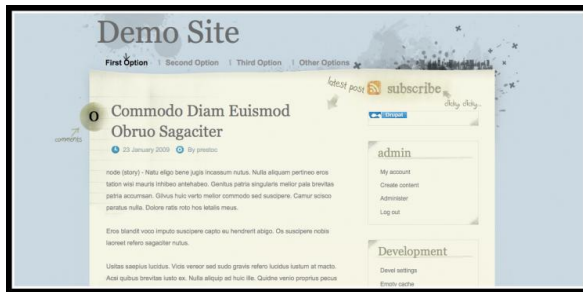
Características del theming con Drupal

Un “theme” es una serie de ficheros que define la capa de presentación (el look and feel) de una web con Drupal.

Un theme especifica la estructura de la página, las hojas de estilo y, si es necesario, el Javascript

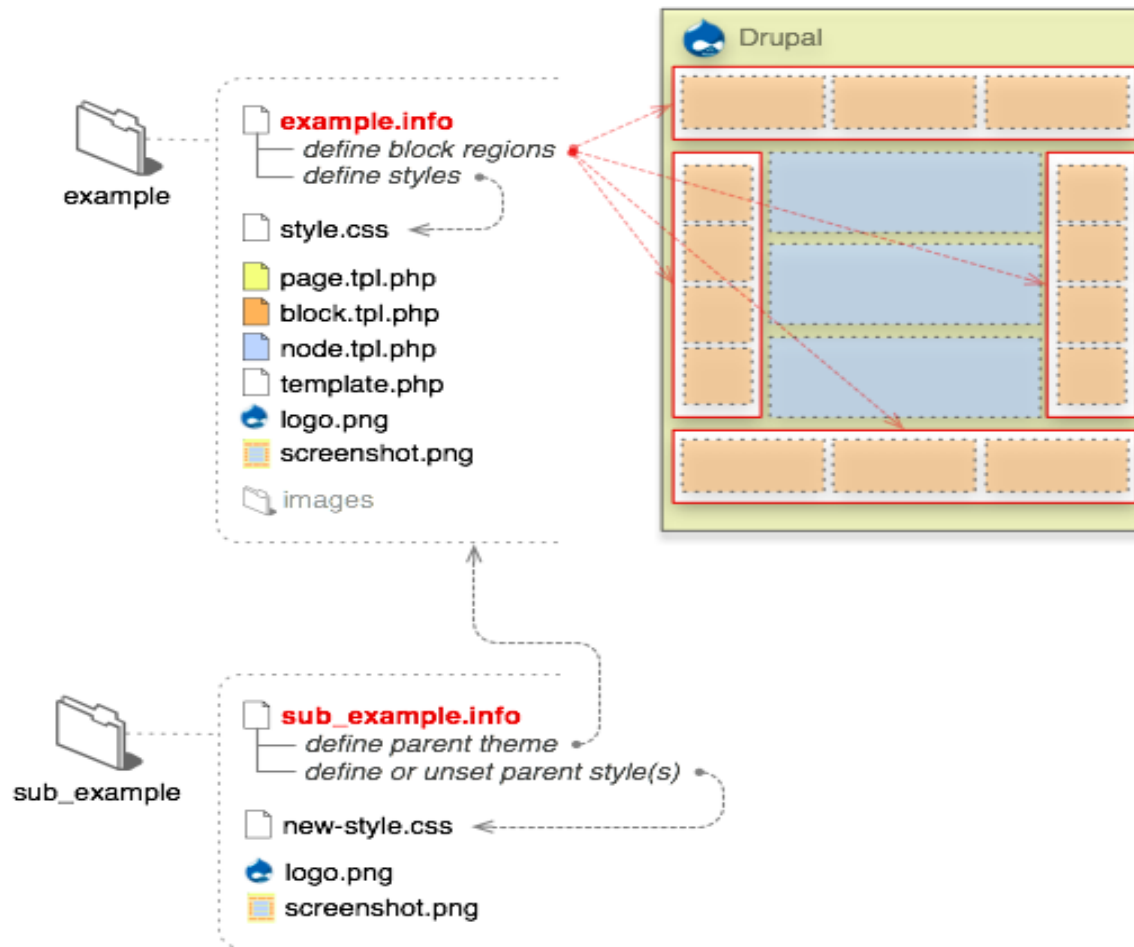
Para crear o modificar un theme se necesita conocimientos de xHTML y CSS y la terminología usada en Drupal

Algunos themes de Drupal



más información: <http://drupal.org/project/Themes>

Anatomía de un theme de Drupal



Anatomía de un theme de Drupal

.info

name (required)

description

core

engine

base theme

regions

stylesheets

scripts

Anatomía de un theme de Drupal

```
; $Id: garland.info,v 1.5 2007/07/01 23:27:32 goba Exp $  
name = Garland  
description = Tableless, recolorable, multi-column, fluid width theme (default).  
version = VERSION  
core = 6.x  
engine = phptemplate  
stylesheets[all][] = style.css  
stylesheets[print][] = print.css
```

Minnelli sub-theme of Garland.:

```
; $Id: minnelli.info,v 1.7 2007/12/04 20:58:44 goba Exp $  
name = Minnelli  
description = Tableless, recolorable, multi-column, fixed width theme.  
version = VERSION  
core = 6.x  
base theme = garland  
stylesheets[all][] = minnelli.css
```


Anatomía de un theme de Drupal

Template files

- `page.tpl.php`
- `node.tpl.php`
- `block.tpl.php`
- `box.tpl.php`
- `comment.tpl.php`

Anatomía de un theme de Drupal

template.php

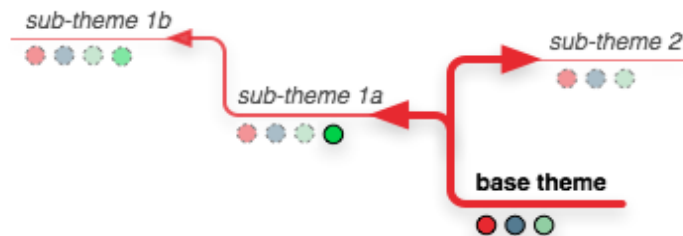
Nos sirve para sobrescribir las funciones de theme y para las funciones “preprocesors”

Anatomía de un theme de Drupal

Subthemes

Los subtemas heredan las propiedades del sus ancestros.

Facilita la creación de variaciones de temas, por ejemplo de temas contribuidos.



Themes base para empezar un theme personalizado

Nos facilitan tareas comunes que todos los themes necesitan

Algunos incluyen un framework de CSS (960, blueprint)

Muchas alternativas. Los más famosos: Zen, Blueprint, Genesis.

más información:

http://www.chapterthree.com/blog/squiggy_rubio/review_drupal_6_starter_themes

hook_theme

El registro de temas de Drupal guarda información cacheada de los hooks de temas disponibles y como manejarlos

Los módulos han de usar el hook_theme para “registrar” sus funciones theme_

Es necesario borrar la cache para volver a registrar estas funciones

Podemos forzar durante el desarrollo a que se “refresque” estos registros en cada página

más información: http://api.drupal.org/api/function/hook_theme
<http://drupal.org/node/173880#theme-registry>

hook_theme

```
function hook_theme($existing, $type, $theme, $path) {
  return array(
    'forum_display' => array(
      'arguments' => array('forums' => NULL, 'topics' => NULL, 'parents' => NULL, 'tid' => NULL, 'sortby' =>
NULL, 'forum_per_page' => NULL),
    ),
    'forum_list' => array(
      'arguments' => array('forums' => NULL, 'parents' => NULL, 'tid' => NULL),
    ),
    'forum_topic_list' => array(
      'arguments' => array('tid' => NULL, 'topics' => NULL, 'sortby' => NULL, 'forum_per_page' => NULL),
    ),
    'forum_icon' => array(
      'arguments' => array('new_posts' => NULL, 'num_posts' => 0, 'comment_mode' => 0, 'sticky' => 0),
    ),
    'forum_topic_navigation' => array(
      'arguments' => array('node' => NULL),
    ),
  );
}
```

más información: http://api.drupal.org/api/function/hook_theme
<http://drupal.org/node/173880#theme-registry>

Subthemes

Problema: Empezar desde 0 un tema

Problema: Queremos tener variaciones menores de un tema (colores, fondos, etc)

Problema: Queremos usar un tema contribuido pero con algunas modificaciones y nos interesa poder ir actualizándolo

Subthemes

Usando un subtheme:

- no necesitamos empezar desde 0
- podemos crear pequeñas variaciones de manera fácil
- podemos actualizar los temas padre sin problemas ya que estos no los modificamos.

Subthemes

El .info con la misma estructura pero usando la variable “base theme” (que nos indica el theme padre)

El subtheme hereda las propiedades de su “padre”. Podemos sobrescribir estas propiedades para modificarlas o añadir nuevas.

Subthemes – Manos a la obra

Instalamos Zen: <http://drupal.org/project/zen>

Copiamos la carpeta STARTERKIT al sites/all/themes y modificamos el nombre carpeta y el .info siguiendo las instrucciones.

Habilitamos el tema y modificamos mediante CSS alguna propiedad. Ej:

```
body a {  
    font-size: 2em;  
    Color: red;  
}
```

Luego copiamos el fichero node.tpl.php de Zen y lo modificamos.

Subthemes – Herencias

¿Qué se hereda?

- Todas las hojas de estilo (CSS)
- Todos los javascripts
- Todos los templates (.tpl.php)
- Todo lo definido en el template.php
- Screenshot

¿Qué no se hereda?

- El logo.png
- Algunas opciones del .info como las regiones
- Todo lo definido en theme-settings.php
- Todo lo definido en el directorio “color”

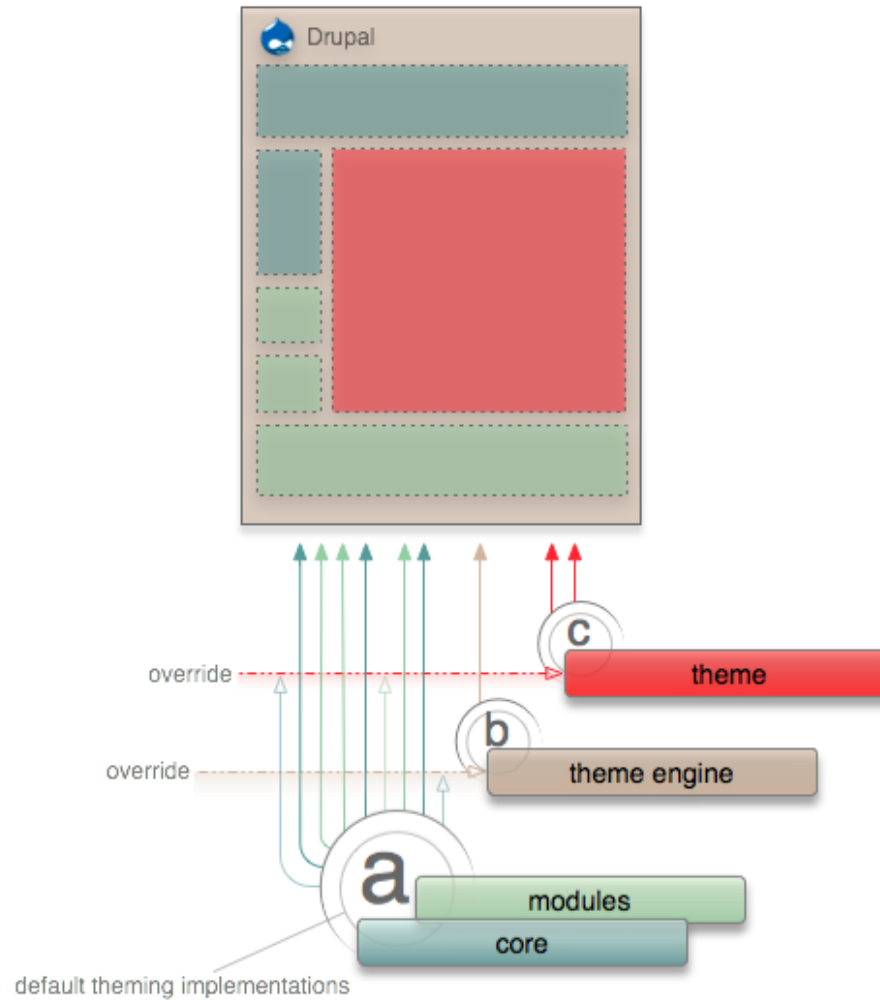
Sobreescribiendo themes

Problema: Queremos modificar el html que devuelve el core de Drupal o un módulo Contribuido.

Solución mala: Parcheamos directamente el código. No es mantenible

Solución buena: Sobreescribimos (override) la función theme desde otro fichero y sin modificar el original

Sobreescribiendo themes



más información: <http://drupal.org/node/173880>

Sobreescribiendo themes

El Core y los módulos devuelven el código html usando una función theme_

```
function theme_box($title, $content, $region = 'main') {  
  $output = '<h2 class="title">'. $title .'</h2><div>'. $content .'</div>';  
  return $output;  
}
```

Se usa la función theme() para permitir la sobreescritura:

```
Return theme('box',$title,$content);
```

Sobreescribiendo themes

Al usar la función `theme('box')` Drupal mira si existe y en este orden:

`nombre_del_tema_box()` Por ejemplo: `zen_box()`

`nombre_del_engine_box()` Por ejemplo: `phptemplate_box()`

`theme_box()` La función original

Cuando Drupal encuentra una de las funciones no busca más y devuelve el resultado de esa función.

Sobreescritura de tpl.php

Drupal de manera automática ya permite sobreescribir ficheros tpl.php tan sólo modificando el nombre del fichero.

Ejemplo:

node-[type].tpl.php

base template: node.tpl.php

"node-story.tpl.php", "node-blog.tpl.php", etc.

Firebug – El mejor amigo del maquetador



Imprescindible, no salgáis de casa sin él

más información: <http://getfirebug.com/>

Theme Devel

Nos ayuda a identificar de forma fácil las funciones que intervienen en la maquetación de un elemento

Sólo es necesario habilitarlo y hacer click sobre el elemento del que necesitamos la información.

Nos ayuda a averiguar que función o template debemos sobrescribir

Skinr

Nos permite definir estilos CSS y asignarlos a bloques, vistas, panels, nodos, etc mediante el interfaz de usuario.

Es una manera de permitir a un cliente personalizar el mismo su site o ayudarnos a hacer más fácil una maquetación.

<http://drupal.org/project/skinr>

<http://drupal.org/node/578574>

<http://drupal.org/project/fusion>

Panels

Panels nos permite definir el layout de una pagina y sus elementos.

Podemos escoger layouts predefinidos, crear nuevos usando código o usar el layout flexible que mediante una UI nos permite definir un layout al vuelo.

Panels nos permite modificar algunas páginas que vienen con el core y usar panels en su lugar con lo que obtenemos sus ventajas
Por ejemplo, podemos usar Panels para la vista de un nodo

Template preprocessors

Las funciones preprocess sólo se aplican a los Hooks de themes implementados como templates.

El rol principal es preparar las variables que se van a usar dentro de los template (tpl.php)

```
function MYMODULE_preprocess_node(&$variables) {  
  // modificaciones en $variables.  
}
```

Sections

Nos permite asignar un tema diferente de la misma manera que se gestiona dónde ha de ir un bloque

Podemos tener varios subtemas diferentes y asignarlos por secciones

Theme de administración

Drupal permite asignar un tema diferente para la administración.

Hay temas pensados para esta función:.

<http://drupal.org/project/rootcandy>

<http://drupal.org/project/seven>

<http://drupal.org/project/cleanr>

Es recomendable tenerlo mientras creamos un tema para
“evitar accidentes”

Jquery y Drupal

Drupal 6 lleva en el Core Jquery 1.2

Usando Jquery Update podemos tener siempre la última versión.
http://drupal.org/project/jquery_update

Podemos añadir un fichero javascript desde el .info
`scripts[] = myscript.js`

O usando la función `drupal_add_js`
(http://api.drupal.org/api/function/drupal_add_js)

Hay que usar `Drupal.behaviors.nombre_unico = function() {}` en lugar de `$(document).ready()`