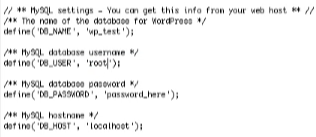
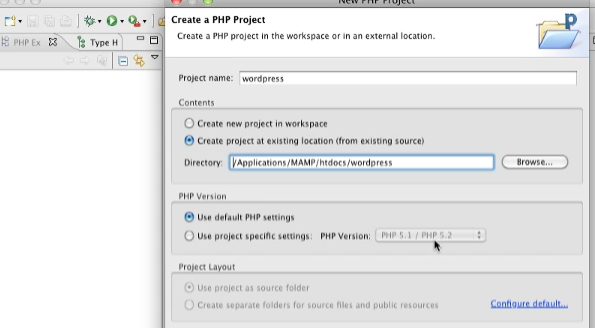
Installing Wordpress:

Create and Edit wp-config

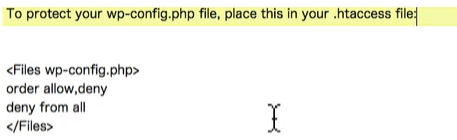
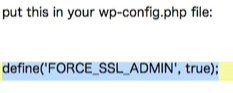




Setup Development Environment

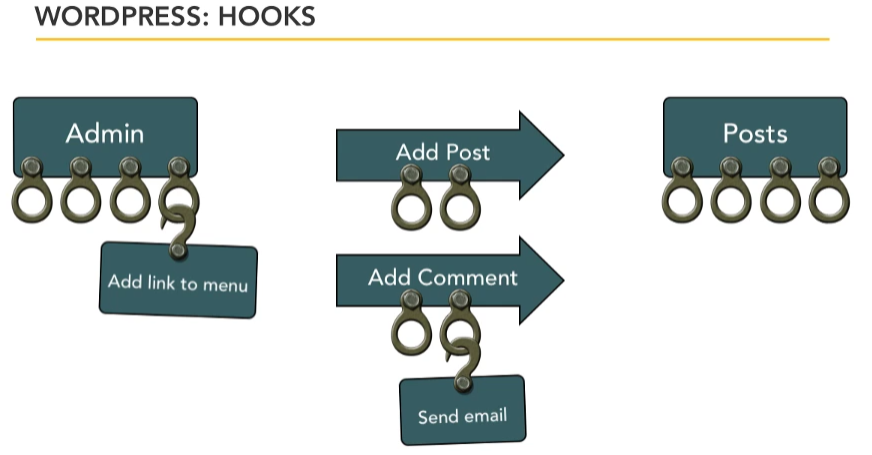


Securing Wordpress

* Use Strong Password
* Don’t Use Admin username
* You can Change the Table Prefix
* Move the wp-config one directory up
  + 
* For SSL Sites
  + 

RSS Feeds: 

0204 Introduction to hooks 

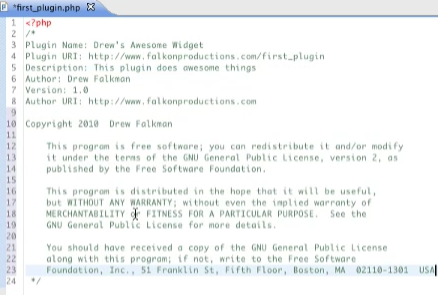






0301 Creating the plugin PHP file(s).mov

Steps:

* 0301 Creating the plugin PHP file(s).movCreate a Unique Plugin Folder
* Create an index.php
  + 

0302 More on hooks Actions and filters.mov

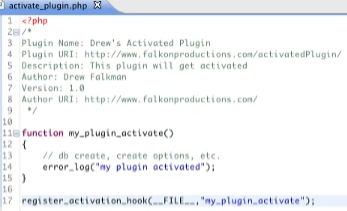
1. [Actions](http://codex.wordpress.org/Plugin_API#Actions): Actions are the hooks that the WordPress core launches at specific points during execution, or when specific events occur. Your plugin can specify that one or more of its PHP functions are executed at these points, using the Action API.
   1. http://codex.wordpress.org/Plugin\_API/Action\_Reference
2. [Filters](http://codex.wordpress.org/Plugin_API#Filters): Filters are the hooks that WordPress launches to modify text of various types before adding it to the database or sending it to the browser screen. Your plugin can specify that one or more of its PHP functions is executed to modify specific types of text at these times, using the Filter API.
   1. http://codex.wordpress.org/Plugin\_API/Filter\_Reference

0303 Installation and activation.mov

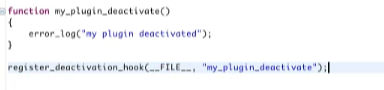
Example Code To Check Wordpress Version



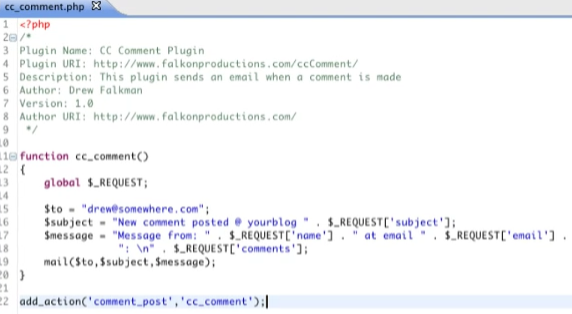
0304 Writing activation code.mov



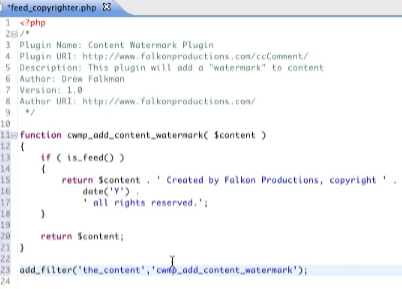




0305 Writing an action.mov



0306 Writing a filter.mov



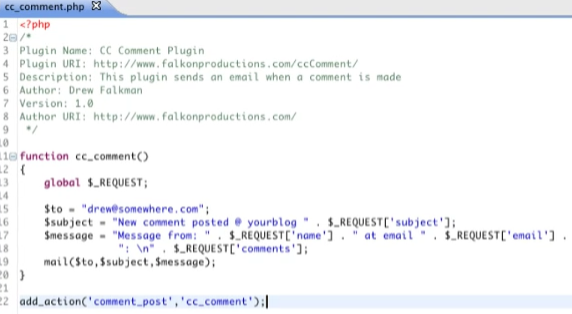
See: 

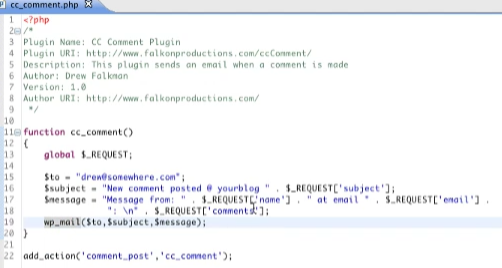
0307 About pluggable functions.mov



For Example: wp\_mail 

Original Code From Previous Exercise



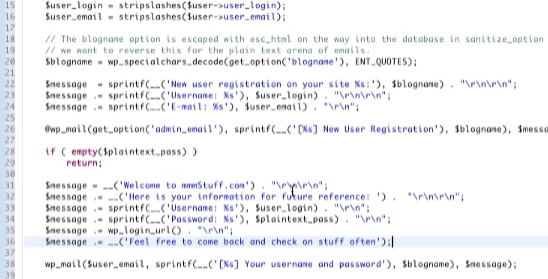
Using the Pluggable PHP 

0308 Writing a pluggable function.mov

Copy the Original Pluggable Functions from Pluggable php file



Modify the Original Pluggable Functions in your plugin



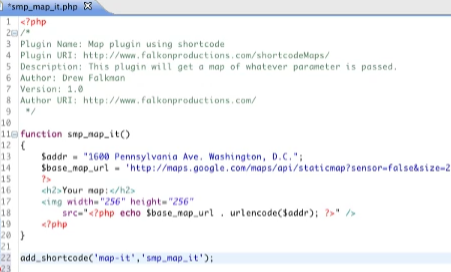
Make sure to check if the function already exist, So there is no conflict



0309 Using template tags.mov

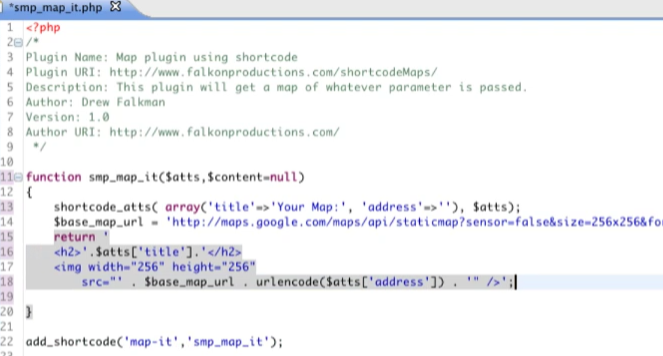


0310 Introducing shortcode.mov





A shortcode that that takes argument





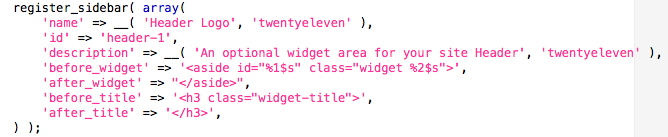
0401 Widgets and the WordPress Widgets SubPanel.mov

0402 Comparing widgets and plugins.mov

Widgets are always a plugin, but a plugin ia not always a widget

0403 Using and customizing built-in widgets.mov

Registering Widgets Area inside functions.php

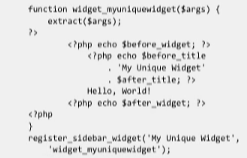
* 
* 
* 

Using the Registered Widgets Area

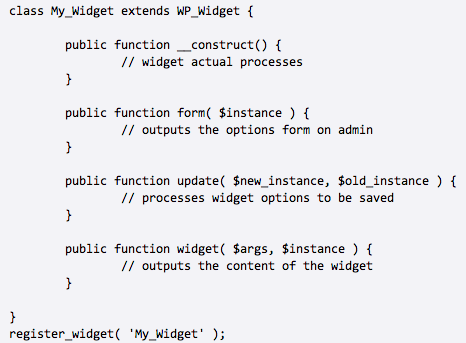
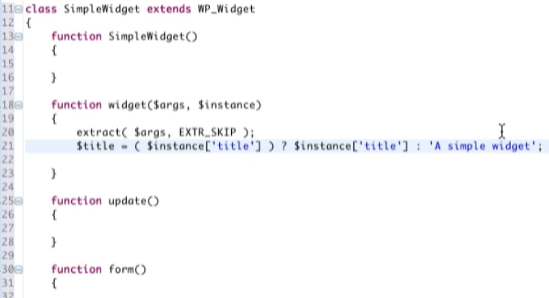
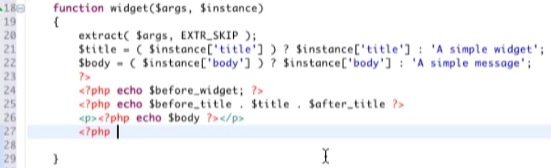
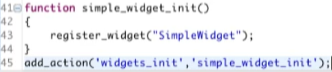
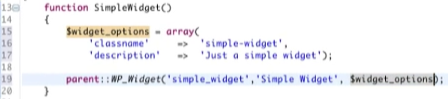
* 

0404 Creating a new widget.mov

Developing Widgets Before wordpress 2.8

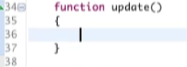
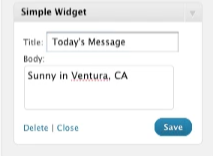
* 
* 

Developing Widgets on or after wordpress 2.8

* <http://codex.wordpress.org/Widgets_API>
* 
* Example:
* 
* The extract() function imports variables into the local symbol table from an array.
  + extract(array,extract\_rules,prefix)
* 
* Register the Widget: widget\_init
  + Make this outside the class
  + 
* View in Plugins Area
* 
* View in Widgets Page
* 
* View when widgets is used
* 
* Setting up the constructor
  + 
  + <http://phpdoc.wordpress.org/trunk/WordPress/Widgets/WP_Widget.html>
  + *WP\_Widget* [**\_\_construct**](http://phpdoc.wordpress.org/trunk/WordPress/Widgets/WP_Widget.html#__construct) ([*string* **$id\_base** = false], *string* **$name**, [*array* **$widget\_options** = array()], [*array* **$control\_options** = array()])
  + View in Widgets Page
  + 

0406 Enabling configuration of widgets.mov



* View in Widgets Page
* 
* Delete
  + 
  + Because all we want is to save the data inputted in Simple Widget
  + If update() is deleted, the plugin will use the wp\_widgets default updated()
  + 
  + 
* Update()
  + This is optional if you want to use data scrubbing
  + It takes 2 argument the old and new instance

0501 Creating an admin interface.mov





Registering Admin Pages For Plugin



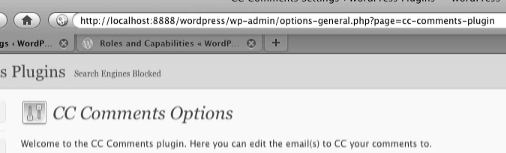
* The last words are cut because it is too long, but all it is, is to call the function name

Calling the registered page

* 

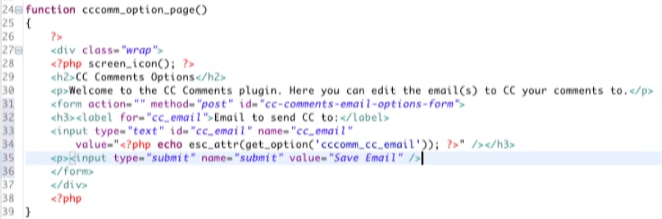
View in Settings





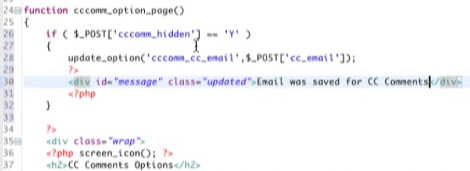
0502 Saving data to the database.mov

Saving Inputted Options Information



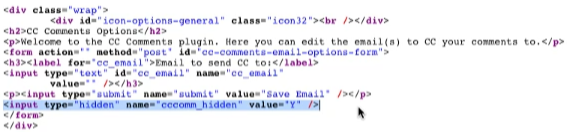
* 

Add This Line of Code

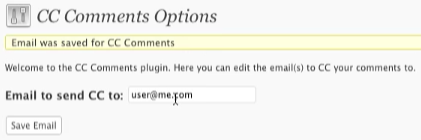




Source View:



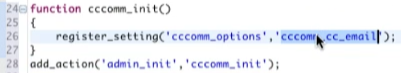
Admin View:

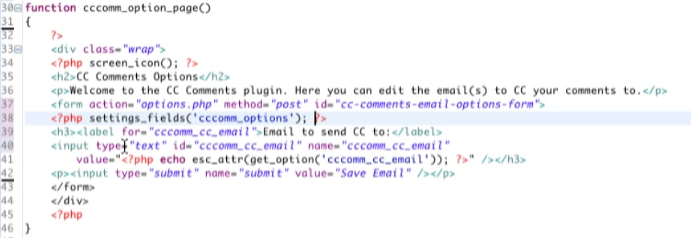


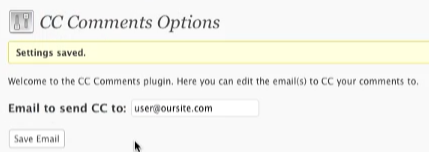
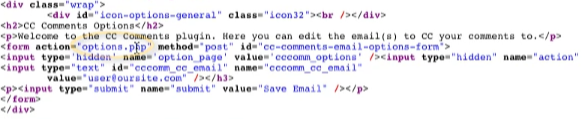
0503 Securing form submission with nonces.mov



0504 Options editing post-WordPress 2.7.mov



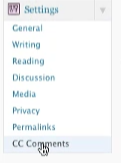


* Admin View
* 
* Source View
* 
* 

0505 Integrating with the WordPress admin menus.mov





* add\_options\_page: Places it under the settings
  + 



* add\_dashboard\_page
  + Adds a menu link on dashboard
  + 

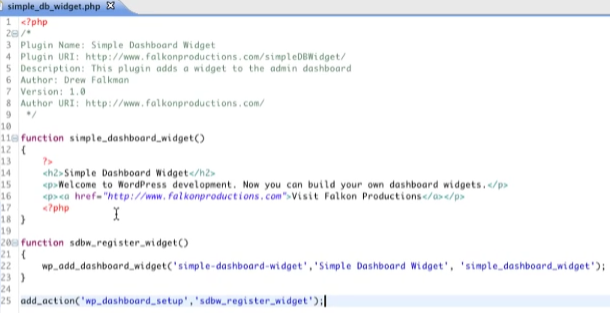


* You can add an Icon and Priority

Other Options for adding the links

* Add\_themes\_page
* Add\_options\_page
* Add\_submenu\_page

0506 WordPress admin dashboard API.mov



Dashboard View:



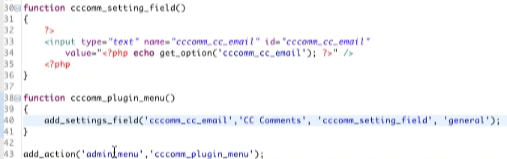
0507 Using existing options and option editing pages in WordPress.mov



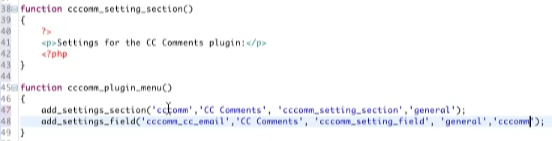
Contains all database settings



Adding Information to All Sttings

* 
  + Make sure the name and id should be the same as the get\_option
* 
* 

To Create Entirely New Section Heading

* 
* 

0508 Using jQuery and AJAX for administration.mov









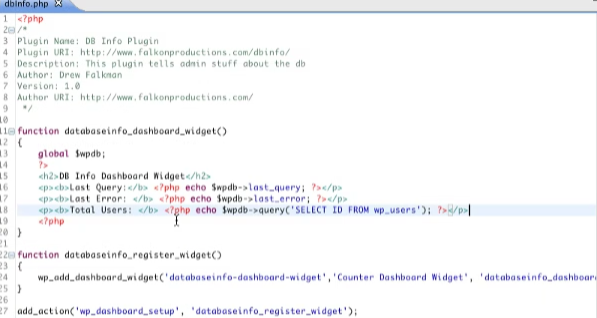
adds admin\_print\_scripts(Javascripts) to options-general.php





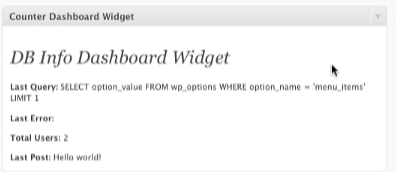
0601 Accessing the WordPress database.mov

Use wpdb

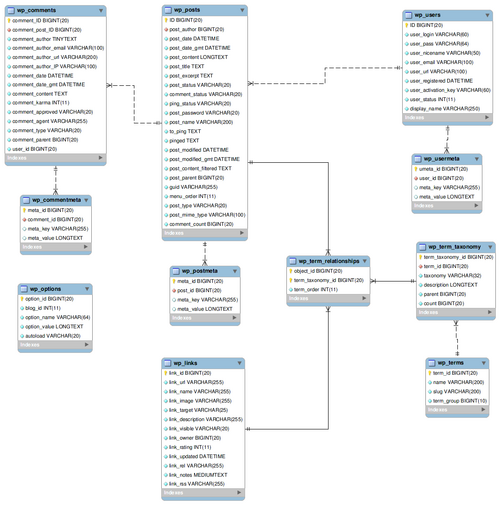




Dashboard View:



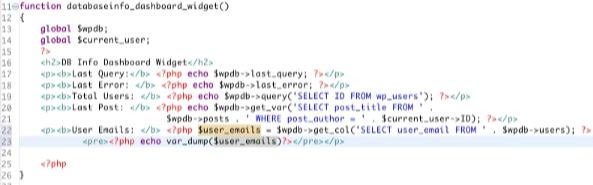
0602 Using the built-in schema.mov



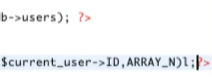
From: <http://codex.wordpress.org/Database_Description>

Use wpdb instead of wp\_posts for compatability

0603 Accessing data using $wpdb.mov





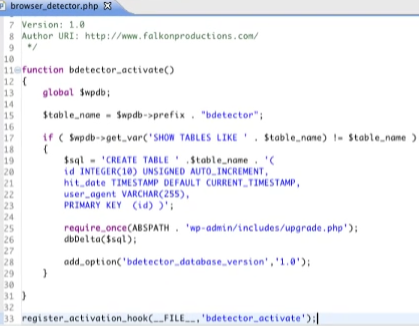


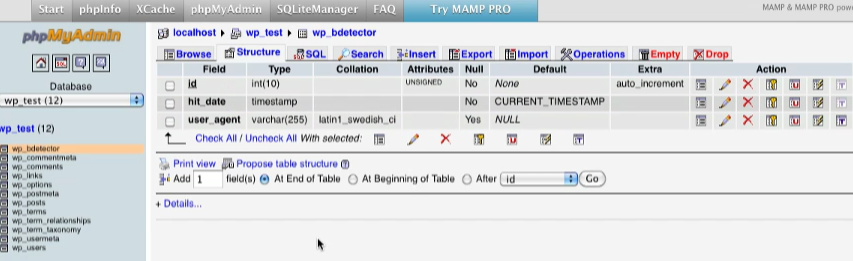
* Array\_N: gets the array number placement
* Array\_A: gets the associative array
* Dashboard View:
* 



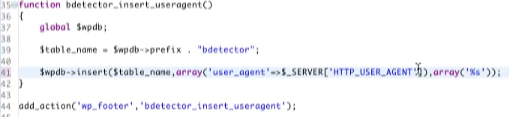


0604 Creating new tables.mov

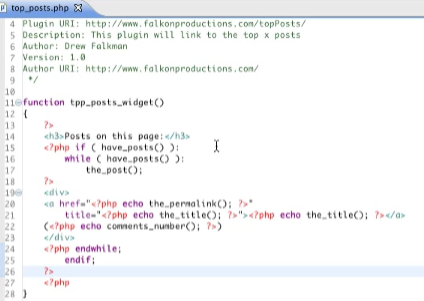




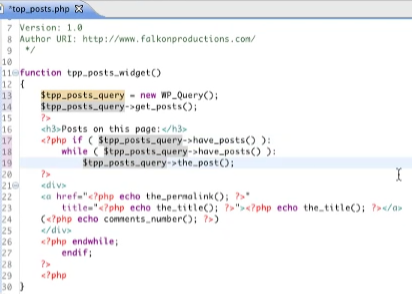
0605 Inserting data.mov

* 
* Read More about wpdb insert and update

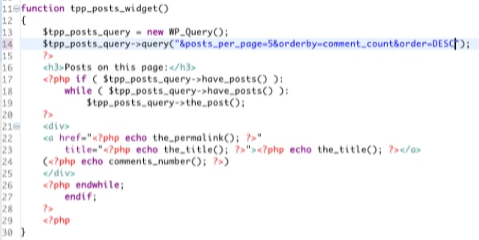
0701 Introducing the Loop.mov



0702 Using WP\_Query().mov

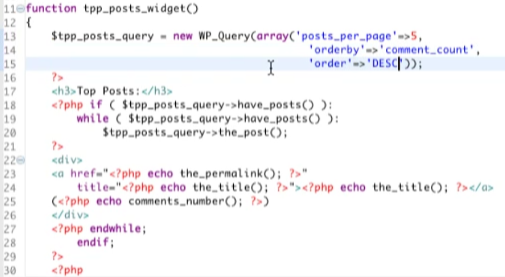


0703 Custom filtering and sticky posts.mov

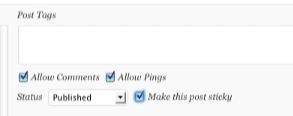
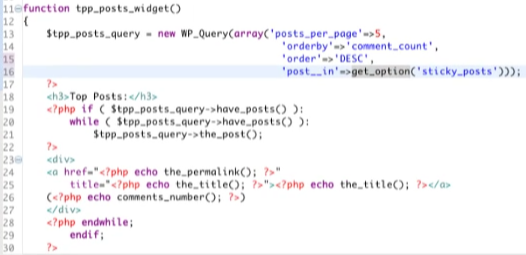


View on Homepage:





To make it sticky

* 
* 
  + Use to query sticky post

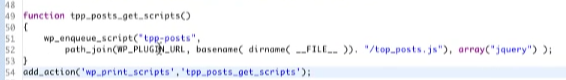
0704 Using jQuery and AJAX for posts and pages.mov

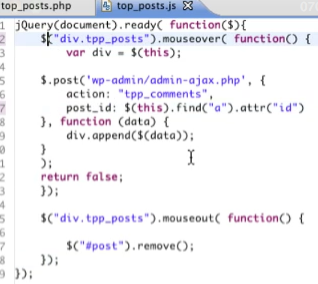
Add Preview when mouse hover



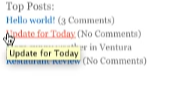








View from front page:

****

0801 Registering and promoting plugins.mov









Other Places to Submit Plugins:

* 
* 

Wordpress Discussion

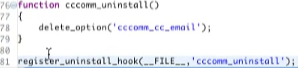
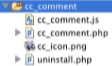
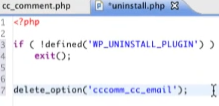






0802 Creating an uninstall function.mov

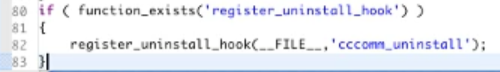
make sure to make a backup of the plugin

* Creating an Uninstall function
  + 
  + This script will uninstall the plugin once it is deleted. However, it will execute the commands above it.
* Creating an Uninstall File
  + 
  + 
  + This script will uninstall the plugin once it is deleted
  + This will run cleanly

0803 Backward compatibility issues.mov



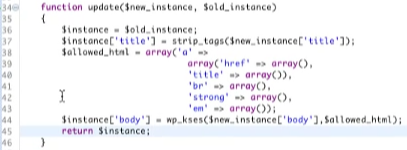
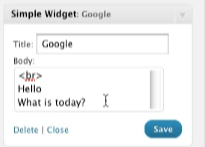




* Check the function if it exist

0901 Understanding security issues.mov

Data Sanitation

* 
  + Forces the input to be string
  + Browser detection plugin
* 
  + prepare
  + Browser detection plugin
* 
  + 
  + Strips the Simple Widget

0902 Internationalizing your plugin.mov







* \_ \_(

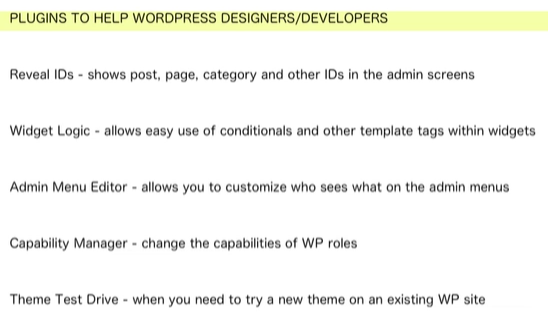


* \_e()
* Creates a POT file

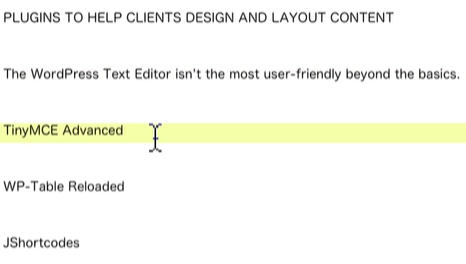
0601 How to Choose Plugins.mp4

* 
* Search Google for best plugin keyword
* Search the authors name or nickname

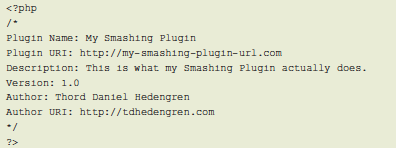
0602 Plugins for Developers.mp4



0603 Plugins for Design and Layout.mp4



Smashing Wordpress Chapter 7

* Plugin Basics
  + Location: wp-content/plugins/
  + Name must be unique
    - Check wordpress.org or in google to make sure your plugin is unique
  + 
    - Header
  + include licensing information
    - [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html)
    - change YEAR and PLUGIN AUTHOR E-MAIL
* Hooks and How to Use Them
  + Syntax: add\_action (**'**hook-name**'**, **'**function-name**'**, X, Y );
    - **X**, is the priority argument (which defaults to 10)
      * The lower the number is, the earlier the function will be executed
    - Y, on the other hand, is the number of arguments your function can accept (defaulting to 1)
  + 2 Types of Hooks
    - Action Hook - triggered by specific events
      * publish\_post()
    - Filter Hook - are functions that WordPress passes data through
      * the\_excerpt()
      * the\_title()
  + Removing Hooks
    - remove\_action(**'**action-hook**'**,**'**the-function**'**)
    - remove\_filter(**'**action-hook**'**,**'**the-function**'**)
* Creating Your Own Template Tags
  + <?php function-name(); ?>
    - Just install the plugin, and you can call the function anywhere in your theme files
      * You don’t need the hook
* The Pluggable Functions - to overwrite parts of the WordPress core functionality, when removing hooks won’t do it
  + Location: wp-includes/pluggable.php
  + you can only do an override once
    - make sure other plugins are not already overriding the functions you want to override
  + to avoid unnecessary errors
    - <?php if (!function\_exists(**'**function-name**'**)); ?>
  + Pluggable functions list: codex.wordpress.org/Pluggable\_Functions