WordPress Settings API CheatSheet — Short Version

Kenneth John Odle — techblog.kjodle.net

Unless otherwise noted, all parameters are required.

v. 1.0 — 2015.07.15

---

add_submenu_page(

$parent_slug,	Which menu item this panel should appear under.
$page_title,	Appears at top of options page in `<h2>` tags.
$menu_title,	Appears in menu in admin sidebar.
$capability,	Users must be able to do this to access these options.
$menu_slug,	Appears in URL
$function	*Optional*. A function that outputs the content for the page.

)

`add_submenu_page()` adds an options page for a plugin under the menu item in `$parent_slug`. If you need to add a top-level options page, use `add_menu_page()` instead. Hook this in the `admin_menu` init.

---

register_setting(

$option_group,	Must match `settings_field($option_group).`
$option_name,	The name of the option stored in the database.
$sanitize_callback	*Optional*. A function to sanitize inputs.

)

`register_setting()` registers a settings ($option_name) in the `wp-options` table.

---

settings_fields(

$option_group	Must match `register_settings($option_group).`

)

`settings_fields()` outputs hidden bits that make the Settings API work. Include it after your opening `<form>` tag.

---

do_settings_sections(

$page	Must match `add_settings_section($page)` and `add_settings_field($page).`

)

`do_settings_sections()` outputs all the input fields created with `add_settings_field()`. Include it after the `settings_fields()` call.

---

add_settings_section(

$id,	For use in HTML tags. Must match `add_settings_field($section).`
$title,	Appears at top of section as an `<h3>.`
$callback,	Function to echo output.
$page	Must match `do_settings_sections($page)` and `add_settings_field($page).`

)

`add_settings_section()` creates a section of settings on your options page. Optional if only a handful of settings are needed.

---

add_settings_field(

$id,	For use in HTML tags
$title,	Displayed to left of option input
$callback,	Function that fills the field with the desired inputs as part of the larger form. The `name` and `id` of the input should match the `$id` given to this function. This information can be contained as an array in the options `$args` parameter.
$page,	Must match `do_settings_sections($page)` (for plugins) and `add_settings_section($page)`.
$section,	*Optional*. Must match `add_settings_section($id)`.
$args	*Optional*. Additional arguments in an `array` that are passed to the `$callback` function. The `'label_for'` key/value pair can be used to format the field title like so: `<label for="value">$title</label>`.

)

`add_settings_field()` creates individual settings within a settings section on your options page. You must register any options used by this function with `register_setting()` or they won't be saved and updated automatically.

The callback function needs to output the appropriate html `input` and fill it with the old value; the saving will then be done behind the scenes.

The html `input` field's `name` attribute must match in `register_setting($option_name)`, and value can be filled using `get_option()`.