

# Exercise 16:

## Simple form altering

---

A big part of Drupal are the forms. For example, content entry forms to add and manage content, Search forms to find content, and settings form to control the numerous parts of our site. Default forms are pretty awesome, but sometimes they just aren't enough. Every now and then, we need to alter a label, default value, or add some libraries for styling or some cool javascript functionality.

Form API and Drupal form alters are still in Drupal 8 are are still incredibly powerful tools to control Drupal. When using a form\_alter for theming, It is best to keep it to simple alterations. Simple alterations include changing labels, adding/removing classes, editing div structure, and adding HTML5 placeholders, like in our following examples. Anything else should really be done in a module form alter.

**For this exercise, make sure the search block is in a region and visible.**

### Simple form altering with `hook_form_alter` function

1. Navigate to your theme root
2. Locate and open your **acme.theme** file, and add the following code.

```
function acme_form_alter(&$form, \Drupal\Core\Form\FormStateInterface $form_state,
  if ($form_id == 'search_block_form') {
    // Add placeholder text
    $form['keys']['#attributes']['placeholder'] = t('Search');
  }
}
```

3. Clear cache, and observe your search block.

### Simple form altering with `hook_form_FORM_ID_alter` function

This function is refined to only effect the form with the matching **FORM\_ID** in the function name.

1. Replace the above function with the following code:

```
function acme_form_search_block_form_alter(&$form, \Drupal\Core\Form\FormStateInterface $form_state) {  
  // Add placeholder text  
  $form['keys']['#attributes']['placeholder'] = t('Search');  
}
```

## Adding a javascript via a library to a form

Let's say we want to leverage one of the libraries from core or contrib when our form is visible.

1. Search and remove any other instances of `kint()` in your theme.
2. View your site as an anonymous user by opening a new browser or private window. Verify that `form.js` is not included.
3. Add the following to the `acme_form_search_block_form_alter` function in **acme.theme**

```
kint(\Drupal::service('library.discovery')->getLibrariesByExtension('core'));
```

4. Search for `form.js` in the the search form provided by kint. You'll see it highlighted under parent `jquery.form`. (The `search_kint` module must be enabled for this to work)
5. Navigate back to your theme root directory and open your `acme.theme` file.
6. Add the following code to your `acme_form_search_block_form_alter` function:

```
// Attach a library from our theme to a form.  
$form['#attached']['library'][] = 'core/drupal.form';
```

7. Clear your caches.

If everything worked, the `form.js` file will now be loaded for anonymous users.

## Questions you may have

- What is the benefit of using `hook_form_FORM_ID_alter` rather than just `hook_form_alter`?
- What else can I do with form alters?
- Can I use `['#attached']` in any function?

Done 😊

