**Form API** <https://drupal.org/node/37775>

For an interactive, data-driven website such as one built with Drupal, collecting and processing user submitted data will be exceptionally important. Most of this data can be captured using a web based form, i.e. An HTML structure with textfields and widgets for selecting different options.

Getting a form on to a webpage is easy, getting the user's responses is easy too, getting them securely is much, much harder. Drupal provides a standard, easy to use, easy to extend and secure way of adding forms to your Drupal website: Form API or FAPI for short.

Creating a builder function to generate a form

Form workflow usually follows these stages:

* **Declaration of a form and its elements in a function**
* **Modification of existing forms with the FAPI hook system**
* **Theming of forms**
* **Form validation and submission**

Forms are represented as arrays in form builder functions. Each item within the $form array on api.drupal.org corresponds either to a form element (an input or other HTML on the rendered form) or an element property (meta-data used by FAPI during the rendering or processing of elements). A property has a key name that begins with a "#", while an element does not.

In this example, we have a module named mymodule and we want to declare a form builder function in it. The form declaration function receives at least two parameters:

* $form - An array of elements and element properties
* &$form\_state - An array containing information about the current state of this form while a user is filling it out and submitting it. Importantly, this includes the values that the user may have entered into the form.

This simple form has a text field for the user name and a submit button.

/\*\*

\* Form constructor for the example form.

\*/

**function mymodule\_example\_form($form, &$form\_state) {**

// Provide a text field.

$form['name'] = array(

'#title' => t('Your full name'),

'#type' => 'textfield',

'#required' => TRUE,

);

// Provide a submit button.

$form['submit'] = array(

'#type' => 'submit',

'#value' => 'Update search',

);

return $form;

}

For more information on types of form elements and the properties they use, see the Form API Reference.

This form can be rendered using the drupal\_render() function:

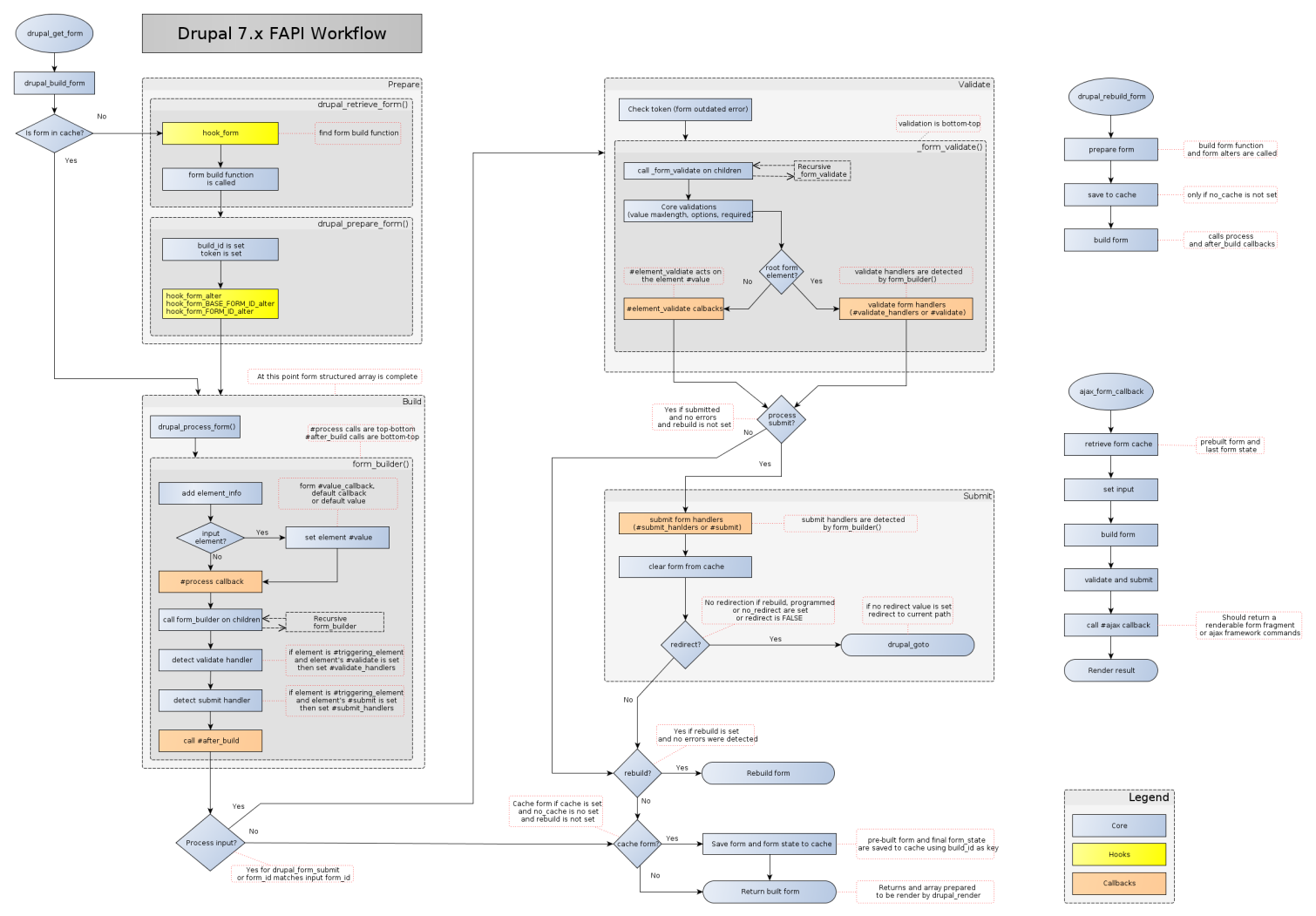
<?php

**$arr = drupal\_get\_form('mymodule\_example\_form');**

**print drupal\_render($arr);**

?>

# Form API Internal Workflow Illustration



# Forms API - Modify Form widgets with hook\_field\_widget\_form\_alter()

Widgets are the parts that make up forms, like textarea widgets, file upload widgets etc.

By altering them you can add, remove or change parts.

You can alter individual form widgets using the following hook:

hook\_field\_widget\_form\_alter()

hook\_field\_widget\_WIDGET\_TYPE\_form\_alter

There are at least two ways to find this WIDGET\_TYPE;

### Checking used widgets using hook\_field\_widget\_form\_alter

### Add the hook\_field\_widget\_form\_alter to your module/theme, like this;

function YOUR\_THEME\_field\_widget\_form\_alter(&$element, &$form\_state, $context) {

echo "\n YOUR\_THEME\_field\_widget\_form\_alter";

echo "\n element: ";

print\_r($element);

echo "\n widget type: ";

print\_r($context['instance']['widget']['type']);

}

Checking the element you can find the widget you are looking for, then the $context['instance']['widget']['type'] contains the WIDGET\_TYPE.

### Checking hook\_field\_widget\_info in the module

You can locate the hook\_field\_widget\_info in the module which widget you want to alter.

Drupal core's File module file.field.inc file contains the following implementation. You can find the type as the keys of the array it returns.

function file\_field\_widget\_info() {

return array(

'file\_generic' => array( // <-- WIDGET\_TYPE

'label' => t('File'),

'field types' => array('file'),

'settings' => array(

'progress\_indicator' => 'throbber',

),

'behaviors' => array(

'multiple values' => FIELD\_BEHAVIOR\_CUSTOM,

'default value' => FIELD\_BEHAVIOR\_NONE,

),

),

);

}

Then you can add the more specific hook\_field\_widget\_WIDGET\_TYPE\_form\_alter().

The following implementation will move the validation description underneath the description, but only when the field name is "field\_design\_files"

function YOUR\_THEME\_field\_widget\_file\_generic\_form\_alter(&$element, &$form\_state, $context) {

if (isset($element['#field\_name']) && $element['#field\_name'] == 'field\_design\_files') {

$upload\_validators = $element[0]['#upload\_validators'];

$element['#description'] .= '' . theme('file\_upload\_help', array('description' => '', 'upload\_validators' => $upload\_validators));

$element['#file\_upload\_description'] = '';

}

}

# Forms API - Modify Forms with hook\_form\_alter()

Drupal 7 forms that modules create can be changed by other modules and even themes with the help of Drupal FAPI hooks system. See Drupal 7 Hooks for a list of Drupal system hooks. The hook that allows to edit the existing forms is hook\_form\_alter. See the hook\_form\_alter() documentation at Drupal API site.

**The syntax is hook\_form\_alter(&$form, &$form\_state, $form\_id).**

In this hook call, **$form is the array of form elements** (see Drupal 7 Form API Overview article). **$form\_state has the keyed array of the state of the form**, that can change on form submission or in multi-step forms. And an important parameter is **$form\_id, which has in it the unique name of the form**. Drupal core will cycle through all the registered forms, passing their initial array, state, and unique id to this hook function.

Example:

For example, you wrote a module mymodule that needs to add a checkbox "I certify that this is my true name" to a small form with a name and a submit button from the Drupal 7 Form API Overview article.

<?php

function mymodule\_form\_alter(&$form, &$form\_state, $form\_id) {

// Check when the right form is passed

if ($form\_id == 'module\_formname') {

// Add the check box

$form['certify'] = array(

'#type' => 'checkbox',

'#title' => t('I certify that this is my true name'),

);

}

}

?>

We can not only add items to a form this way, but also remove existing elements, assign additional attributes to the elements, or edit the attributes of existing elements as well.

Note that property names in **bold** are those that will generally need to be defined when creating this form element. Default values are indicated in parentheses next to property names, if they exist.

### actions

**Description**: **A wrapper element to group one or more buttons in a form.** Use of the 'actions' element as an array key helps to ensure proper styling in themes and to enable other modules to properly alter a form's actions.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [**#attributes**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes) [#children](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#children), [#id](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#id), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [#type](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example:**

<?php  
  $form['actions'] = [array](http://php.net/array)('#type' => 'actions');  
  $form['actions']['submit'] = [array](http://php.net/array)(  
    '#type' => 'submit',  
    '#value' => t('Save'),  
  );  
  $form['actions']['delete'] = [array](http://php.net/array)(  
    '#type' => 'button',  
    '#value' => t('Delete'),  
  );  
  $form['actions']['cancel'] = [array](http://php.net/array)(  
    '#markup' => l(t('Cancel'), 'foo/bar'),  
  );  
?>

### button

**Description**: **Format an action button. When the button is pressed, the form will be submitted to Drupal, where it is validated and rebuilt.** **The submit handler is not invoked.**

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [#button\_type](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button_type) (default: submit), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#executes\_submit\_callback](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#executes_submit_callback) (default: FALSE), [#limit\_validation\_errors](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#limit_validation_errors), [#name](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#name) (default: op), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit_property), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#validate), [**#value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#value), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([node.module](https://api.drupal.org/api/drupal/7.x/search/node.module" \o "Multiple implementations exist.)):

<?php  
$form['preview'] = [array](http://php.net/array)(  
  '#type' => 'button',   
  '#value' => t('Preview'),   
  '#weight' => 19,  
);  
?>

### checkbox

**Description**: Format a checkbox.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled),[#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#field\_prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_prefix), [#field\_suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_suffix), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process),[#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [**#return\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#return_value) (default: 1), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display)(default: after), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([contact.module](https://api.drupal.org/api/drupal/modules%21contact%21contact.module/7.x" \o "Enables the use of personal and site-wide contact forms.)):

<?php  
$form['copy'] = [array](http://php.net/array)(  
  '#type' => 'checkbox',   
  '#title' => [t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Send me a copy.'),  
);  
?>

### checkboxes

**Description**: Format a set of checkboxes. #options is an associative array, where the key is the #return\_value of the checkbox and the value is displayed. The #options array cannot have a 0 key, as it would not be possible to discern checked and unchecked states.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled),[#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [**#options**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#options), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states),[#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [**#tree**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree) (default: TRUE), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([form\_example\_states.inc](http://api.drupal.org/api/examples/form_example--form_example_states.inc)):

<?php  
$form['high\_school']['tests\_taken'] = [array](http://php.net/array)(  
  '#type' => 'checkboxes',  
  '#options' => drupal\_map\_assoc([array](http://php.net/array)(t('SAT'), t('ACT'))),  
  '#title' => t('What standardized tests did you take?'),  
...  
),  
?>

### container

**Description**: Returns HTML to wrap child elements in a container. Surrounds child elements with a <div> and adds attributes such as classes or an HTML id.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [**#attributes**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes) [#children](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#children), [#id](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#id), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [#type](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([field.form.inc](https://api.drupal.org/api/drupal/modules%21field%21field.form.inc/7.x)):

<?php  
  if ($elements) {  
    // Also aid in theming of field widgets by rendering a classified  
    // container.  
    $addition[$field\_name] = [array](http://php.net/array)(  
      '#type' => 'container',  
      '#attributes' => [array](http://php.net/array)(  
        'class' => [array](http://php.net/array)(  
          'field-type-' . drupal\_html\_class($field['type']),  
          'field-name-' . drupal\_html\_class($field\_name),  
          'field-widget-' . drupal\_html\_class($instance['widget']['type']),  
        ),  
      ),  
      '#weight' => $instance['widget']['weight'],  
    );  
  }  
?>

### date

**Description**: Format a date selection box. The #default\_value will be today's date if no value is supplied. The format for the #default\_value and the #return\_value is an array with three elements with the keys: 'year', month', and 'day'. For example, [array](http://php.net/array)('year' => 2007, 'month' => 2, 'day' => 15)

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers),[**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([profile.module](https://api.drupal.org/api/drupal/7.x/search/profile.module" \o "Multiple implementations exist.)):

<?php  
$fields[$category][$field->name] = [array](http://php.net/array)(  
  '#type' => 'date',   
  '#title' => check\_plain($field->title),   
  '#default\_value' => $edit[$field->name],   
  '#description' => \_profile\_form\_explanation($field),   
  '#required' => $field->required  
);  
?>

### fieldset

**Description**: Format a group of form items.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#collapsed**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#collapsed) (default: FALSE), [**#collapsible**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#collapsible) (default: FALSE), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([contact.module](https://api.drupal.org/api/drupal/modules%21contact%21contact.module/7.x" \o "Enables the use of personal and site-wide contact forms.)):

<?php  
$form['contact'] = [array](http://php.net/array)(  
  '#type' => 'fieldset',   
  '#title' => t('Contact settings'),   
  '#weight' => 5,   
  '#collapsible' => TRUE,   
  '#collapsed' => FALSE,  
);  
?>

### file

**Description**: Format a file upload field.

Note: Drupal will add the code enctype="multipart/form-data", required by browsers to handle files, so it's not necessary to include it yourself.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#array\_parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#array_parents), [#attached](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attached), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled),[#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/required), [**#size**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#size) (default: 60), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([file.module](https://api.drupal.org/api/drupal/modules%21file%21file.module/7.x" \o "Defines a \"managed_file\" Form API field and a \"file\" field for Field module.)):

<?php  
$element['upload'] = [array](http://php.net/array)(  
    '#name' => 'files[' . implode('\_', $element['#parents']) . ']',  
    '#type' => 'file',  
    '#title' => t('Choose a file'),  
    '#title\_display' => 'invisible',  
    '#size' => 22,  
    '#theme\_wrappers' => [array](http://php.net/array)(),  
    '#weight' => -10,  
  );  
?>

### form

**Description**: A form containing form elements

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#action](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#action) (default: [request\_uri](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/request_uri/7.x)()), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [#method](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#method) (default: 'post'), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit_property), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [#validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#validate)

**Usage example**:

N/A

### hidden

**Description**: Store data in a hidden form field.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#default\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [**#value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#value), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([block.module](https://api.drupal.org/api/drupal/7.x/search/block.module" \o "Multiple implementations exist.)):

<?php  
$form['bid'] = [array](http://php.net/array)('#type' => 'hidden', '#value' => $bid);  
?>

Note that if you plan to use JavaScript to change your hidden element's value, you will need to use the #default\_valueproperty rather than #value.

### image\_button

**Description**: Format a form submit button with an image.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [#button\_type](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button_type) (default: 'submit'), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#executes\_submit\_callback](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#executes_submit_callback) (default: TRUE), [#limit\_validation\_errors](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#limit_validation_errors), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render),[#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#return\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#return_value) (default: TRUE), [#src](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#src), [#submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit_property), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#validate), [**#value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#value), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

### item

**Description**: Generate a display-only form element allowing for an optional title and description.

Note: since this is a read-only field, setting the [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required) property will do nothing except theme the form element to look as if it were actually required (i.e. by placing a red star next to the [#title](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title)).

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#markup](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#markup-attribute), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix),[#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree),[**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([contact.module](https://api.drupal.org/api/drupal/modules%21contact%21contact.module/7.x" \o "Enables the use of personal and site-wide contact forms.)):

<?php  
$form['from'] = [array](http://php.net/array)(  
  '#type' => 'item',  
  '#title' => [t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('From'),  
  '#markup' => $user->name .' &lt;'. $user->mail .'&gt;',  
);  
?>

### machine\_name

**Description**: Format a single-line text field to store a unique machine-readable name.

Provides a form element to enter a machine name, which is validated to ensure that the name is unique and does not contain disallowed characters. All disallowed characters are replaced with a replacement character via JavaScript.

The element validation checks that the submitted value

* does not contain the replacement character only
* does not contain disallowed characters
* is unique; i.e., does not already exist
* does not exceed the maximum length (via [#maxlength](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#maxlength))
* cannot be changed after creation (via [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled))

**Non-standard form element properties**:

* #machine\_name: (array)
  + exists: A function name to invoke for checking whether a submitted machine name value already exists. The submitted value is passed as argument. In most cases, an existing API or menu argument function can be re-used. The callback is only invoked, if the submitted value differs from the element's #default\_value
  + source: (optional) The #array\_parents of the form element containing the human-readable name (i.e., as contained in the $form structure) to use as source for the machine name. Defaults to [array](http://php.net/array)('name').
  + label: (optional) A text to display as label for the machine name value after the human-readable name form element. Defaults to "Machine name".
  + replace\_pattern: (optional) A regular expression (without delimiters) matching disallowed characters in the machine name. Defaults to '[^a-z0-9\_]+'.
  + replace: (optional) A character to replace disallowed characters in the machine name via JavaScript. Defaults to '\_' (underscore). When using a different character, 'replace\_pattern' needs to be set accordingly.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [#autocomplete\_path](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#autocomplete_path) (default: FALSE), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value),[#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description) (default: 'A unique machine-readable name. Can only contain lowercase letters, numbers, and underscores.'), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#field\_prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_prefix), [#field\_suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_suffix), [**#maxlength**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#maxlength) (default: 64), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents),[#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required) (default: TRUE), [**#size**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#size) (default: 60), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix),[#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme) (default: 'textfield'), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers) (default: 'form\_element'), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title) (default: 'Machine-readable name'), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display) [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([taxonomy.admin.inc](https://api.drupal.org/api/drupal/7.x/search/taxonomy.admin.inc)):

<?php  
$form['machine\_name'] = [array](http://php.net/array)(  
  '#type' => 'machine\_name',  
  '#default\_value' => $vocabulary->machine\_name,  
  '#maxlength' => 21,  
  '#machine\_name' => [array](http://php.net/array)(  
    'exists' => 'menu\_edit\_menu\_name\_exists',  
  ),  
);  
?>

**Usage example** ([menu.admin.inc](https://api.drupal.org/api/drupal/7.x/search/menu.admin.inc)):

<?php  
$form['menu\_name'] = [array](http://php.net/array)(  
  '#type' => 'machine\_name',  
  '#title' => t('Menu name'),  
  '#default\_value' => $menu['menu\_name'],  
  '#maxlength' => [MENU\_MAX\_MENU\_NAME\_LENGTH\_UI](https://api.drupal.org/api/drupal/modules%21menu%21menu.module/constant/MENU_MAX_MENU_NAME_LENGTH_UI/7.x),  
  '#description' => t('A unique name to construct the URL for the menu. It must only contain lowercase letters, numbers and hyphens.'),  
  '#machine\_name' => [array](http://php.net/array)(  
    'exists' => 'menu\_edit\_menu\_name\_exists',  
    'source' => [array](http://php.net/array)('title'),  
    'label' => t('URL path'),  
    'replace\_pattern' => '[^a-z0-9-]+',  
    'replace' => '-',  
  ),  
  // A menu's machine name cannot be changed.  
  '#disabled' => !empty($menu['old\_name']) || isset($system\_menus[$menu['menu\_name']]),  
);  
?>

### managed\_file

**Description**: Provides a complete ajax/progress aware widget for uploading a file and saving it to the {file\_managed} table.

By default a simple upload button is provided for choosing the file to upload. Once a file has been chosen it can be explicitly uploaded via the "Upload" button. The upload is via AJAX and a progress meter is displayed. The form's validate and submit handlers receive a file object ID in $form\_state['values'] that represents the ID of the new file in the {file\_managed} table.

The #managed\_file element is expanded into a set of FAPI elements including two '#submit' buttons one for Upload one for Remove, a '#file' element and a handful of '#hidden' and '#markup' elements to handle progress indication and displaying of already uploaded files.

**Note:** New files are uploaded with a status of 0 and are treated as temporary files which are removed after 6 hours via cron. Your module is responsible for changing the $file objects status to [FILE\_STATUS\_PERMANENT](https://api.drupal.org/api/drupal/includes%21file.inc/constant/FILE_STATUS_PERMANENT/7.x) and saving the new status to the database. Something like the following within your submit handler should do the trick.

<?php  
// Load the file via file.fid.  
$file = file\_load($form\_state['values']['my\_file\_field']);  
// Change status to permanent.  
$file->status = [FILE\_STATUS\_PERMANENT](https://api.drupal.org/api/drupal/includes%21file.inc/constant/FILE_STATUS_PERMANENT/7.x);  
// Save.  
file\_save($file);  
// Record that the module (in this example, user module) is using the file.   
file\_usage\_add($file, 'user', 'user', $account->uid);   
?>

(Without the call to file\_usage\_add, file\_managed\_file\_validate produces an error upon saving the form, saying that the uploaded file may not be referenced.)

Once a file has been uploaded the file object's fid can be used as the #default\_value for the form element. This will display an icon, a link to the file, and a remove button.

Clicking the remove button sets the value of the field to 0 and your module is responsible for actually removing the file from the files table and the file system using [file\_delete](https://api.drupal.org/api/drupal/includes%21file.inc/function/file_delete/7.x" \o "Deletes a file and its database record.)().

**Non-standard form element properties**:

* #progress\_indicator: options are 'none', 'bar', and 'throbber', default is 'throbber'.
* #progress\_message: (string) Progress message to display along with progress meter while a file is being uploaded. Defaults to NULL.
* #upload\_validators: (array) an array of callback functions to perform validation of uploaded files.
* #upload\_location (string) location on server where uploaded files should be stored. e.g.) 'public://files/my\_files\_director'

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#array\_parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#array_parents), [#attached](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attached), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme),[#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#title](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weight)

**Usage example** ([image\_example.pages.inc](http://api.drupal.org/api/drupal/developer--examples--image_example--image_example.pages.inc).):

<?php  
// Use the #managed\_file FAPI element to upload an image file.  
$form['image\_example\_image\_fid'] = [array](http://php.net/array)(  
  '#title' => t('Image'),  
  '#type' => 'managed\_file',  
  '#description' => t('The uploaded image will be displayed on this page using the image style choosen below.'),  
  '#default\_value' => variable\_get('image\_example\_image\_fid', ''),  
  '#upload\_location' => 'public://image\_example\_images/',  
);  
?>

### markup

**Description**: Generate generic markup for display inside forms. Note that there is **no need to declare a form element as #type = 'markup'**, as this is the default type.

Note: if you use markup, if your content is not wrapped in tags (generally <p> or <div>), your content will fall outside of collapsed fieldsets.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#markup](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#markup-attribute), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [#type](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([contact.module](https://api.drupal.org/api/drupal/modules%21contact%21contact.module/7.x" \o "Enables the use of personal and site-wide contact forms.)):

<?php  
$form['contact\_information'] = [array](http://php.net/array)(  
  '#markup' => variable\_get('contact\_form\_information', t('You can leave us a message using the contact form below.')),  
);  
?>

### password

**Description**: Format a single-line text field that does not display its contents visibly.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#field\_prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_prefix), [#field\_suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_suffix), [**#maxlength**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#maxlength) (default: 128), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [**#size**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#size) (default: 60), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([user.module](https://api.drupal.org/api/drupal/7.x/search/user.module" \o "Multiple implementations exist.)):

<?php  
$form['pass'] = [array](http://php.net/array)(  
  '#type' => 'password',   
  '#title' => t('Password'),   
  '#maxlength' => 64,   
  '#size' => 15,  
);  
?>

### password\_confirm

**Description**: Format a pair of [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password) fields, which do not validate unless the two entered passwords match.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#field\_prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_prefix), [#field\_suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_suffix),[#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [**#size**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#size) (default: 60), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([user.module](https://api.drupal.org/api/drupal/7.x/search/user.module" \o "Multiple implementations exist.)):

<?php  
$form['pass'] = [array](http://php.net/array)(  
  '#type' => 'password\_confirm',   
  '#title' => t('Password'),   
  '#size' => 25,  
);  
?>

### radio

**Description**: Format a radio button.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled),[#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#field\_prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_prefix), [#field\_suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_suffix), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process),[#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [#return\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#return_value), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display) (default: after),[#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example**:

N/A

### radios

**Description**: Format a set of radio buttons.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled),[#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [**#options**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#options), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states),[#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([poll.module](https://api.drupal.org/api/drupal/7.x/search/poll.module" \o "Multiple implementations exist.)):

<?php  
  $active = [array](http://php.net/array)(0 => t('Closed'), 1 => t('Active'));  
...  
  $form['settings']['active'] = [array](http://php.net/array)(  
    '#type' => 'radios',  
    '#title' => t('Poll status'),  
    '#default\_value' => [isset](http://php.net/isset)($node->active) ? $node->active : 1,  
    '#options' => $active,  
    '#description' => t('When a poll is closed, visitors can no longer vote for it.'),  
    '#access' => $admin,  
  );  
?>

### select

**Description**: Format a drop-down menu or scrolling selection box. See [#empty\_option](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#empty_option) and [#empty\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#empty_value) for an explanation of various settings for a select element, including behavior if #required is TRUE or FALSE.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled),[#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [**#empty\_option**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#empty_option), [**#empty\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#empty_value), [#field\_prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_prefix), [#field\_suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_suffix), [**#multiple**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#multiple), [**#options**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#options), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents),[#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [#size](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#size), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers),[**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([contact.admin.inc](https://api.drupal.org/api/drupal/modules%21contact%21contact.admin.inc/7.x)):

<?php  
  $form['selected'] = [array](http://php.net/array)(  
    '#type' => 'select',  
    '#title' => t('Selected'),  
    '#options' => [array](http://php.net/array)(  
      0 => t('No'),  
      1 => t('Yes'),  
    ),  
    '#default\_value' => $category['selected'],  
    '#description' => t('Set this to <em>Yes</em> if you would like this category to be selected by default.'),  
  );  
?>

### submit

**Description**: Format a form submit button.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [#button\_type](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button_type) (default: 'submit'), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#executes\_submit\_callback](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#executes_submit_callback) (default: TRUE), [#limit\_validation\_errors](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#limit_validation_errors), [#name](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#name) (default: 'op'), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit_property), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#validate), [**#value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#value), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([locale.module](https://api.drupal.org/api/drupal/7.x/search/locale.module" \o "Multiple implementations exist.)):

<?php  
$form['submit'] = [array](http://php.net/array)('#type' => 'submit', '#value' => t('Import'));  
?>

### tableselect

**Description**: A table created with a far left column of radios or checkboxes. Build the table headings and columns with the #headers property, and the rows with the #options property. See [Adding checkboxes to a table](http://drupal.org/node/945102#drupal7) for a full explanation.

Other settings:

* To set accessibility tags for the radios or checkboxes, build one of the cells in the #options property using the 'title' => [array](http://php.net/array)( 'data' => [array](http://php.net/array)(#title => 'mytitle'))) construction, as in the $options variable in the usage example below. Drupal will create invisible label tags for the left column based on this value.
* To disable the default "check all" button for the checkboxes, set #js\_select property to FALSE. This is FALSE by default for radios.
* Setting #multiple to TRUE will give you checkboxes instead of radios.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#empty](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#empty), [#header](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#header_prop), [#js\_select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#js_select), [**#multiple**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#multiple), [**#options**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#options), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([node.admin.inc](https://api.drupal.org/api/drupal/modules%21node%21node.admin.inc/7.x)):

<?php  
  // Build the sortable table header.  
  $header = [array](http://php.net/array)(  
    'title' => [array](http://php.net/array)('data' => t('Title'), 'field' => 'n.title'),  
    'type' => [array](http://php.net/array)('data' => t('Type'), 'field' => 'n.type'),  
    'author' => t('Author'),  
    'status' => [array](http://php.net/array)('data' => t('Status'), 'field' => 'n.status'),  
    'changed' => [array](http://php.net/array)('data' => t('Updated'), 'field' => 'n.changed', 'sort' => 'desc')  
  );  
...  
  //Get the node data.  
  $nids = $query  
    ->fields('n',array('nid'))  
    ->limit(50)  
    ->orderByHeader($header)  
    ->execute()  
    ->fetchCol();  
  $nodes = node\_load\_multiple($nids);  
...  
  //Build the rows.  
  $options = [array](http://php.net/array)();  
  foreach ($nodes as $node) {  
...  
    $options[$node->nid] = [array](http://php.net/array)(  
      'title' => [array](http://php.net/array)(  
        'data' => [array](http://php.net/array)(  
          '#type' => 'link',  
          '#title' => $node->title,  
          '#href' => 'node/' . $node->nid,  
          '#options' => $l\_options,  
          '#suffix' => ' ' . theme('mark', [array](http://php.net/array)('type' => node\_mark($node->nid, $node->changed))),  
        ),  
      ),  
      'type' => check\_plain(node\_type\_get\_name($node)),  
      'author' => theme('username', [array](http://php.net/array)('account' => $node)),  
      'status' => $node->status ? t('published') : t('not published'),  
      'changed' => format\_date($node->changed, 'short'),  
    );  
   //For simplicity, this example omits the code to set the operations column.  
...  
  //Build the tableselect.  
  $form['nodes'] = [array](http://php.net/array)(  
      '#type' => 'tableselect',  
      '#header' => $header,  
      '#options' => $options,  
      '#empty' => t('No content available.'),  
    );  
?>

### text\_format

**Description**: A text-format-enabled version of a textarea.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#cols**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#cols) (default: 60), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [#resizable](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#resizable)(default: TRUE), [**#rows**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#rows) (default: 5), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display),[#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree),[**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Non-standard Properties**:

* #format: the format to apply. If you want to use the default format, set this property to NULL in your form constructor function, and the filter system will handle the rest.
* #base\_type (optional): defaults to 'textarea'. This makes it possible to also attach the text format selector to other form element types, such as textfields.

**Usage example** ([taxonomy.admin.inc](https://api.drupal.org/api/drupal/7.x/search/taxonomy.admin.inc)):

<?php  
  $form['description'] = [array](http://php.net/array)(  
    '#type' => 'text\_format',  
    '#title' => t('Description'),  
    '#default\_value' => $term->description,  
    '#format' => $term->format,  
    '#weight' => 0,  
  );  
?>

### textarea

**Description**: Format a multiple-line text field.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#cols**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#cols) (default: 60), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#field\_prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_prefix), [#field\_suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_suffix), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [#resizable](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#resizable) (default: TRUE), [**#rows**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#rows) (default: 5), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([comment.module](https://api.drupal.org/api/drupal/7.x/search/comment.module" \o "Multiple implementations exist.)):

<?php  
$form['keywords'] = [array](http://php.net/array)(  
'#title' => t('Keywords'),  
'#type' => 'textarea',  
'#description' => t ('The comment will be unpublished if it contains any of the phrases above. Use a case-sensitive, comma-separated list of phrases. Example: funny, bungee jumping, "Company, Inc."'),  
'#default\_value' => [isset](http://php.net/isset)( $context['keywords']) ? drupal\_implode\_tags($context['keywords']) : '',  
);  
?>

### textfield

**Description**: Format a single-line text field.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#ajax](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [#autocomplete\_path](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#autocomplete_path) (default: FALSE), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value),[#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#field\_prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_prefix), [#field\_suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#field_suffix), [**#maxlength**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#maxlength) (default: 128), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [**#size**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#size) (default: 60), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#text_format),[#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([forum.module](https://api.drupal.org/api/drupal/7.x/search/forum.module" \o "Multiple implementations exist.)):

<?php  
$form['title'] = [array](http://php.net/array)(  
  '#type' => 'textfield',   
  '#title' => t('Subject'),   
  '#default\_value' => $node->title,   
  '#size' => 60,   
  '#maxlength' => 128,   
  '#required' => TRUE,  
);  
?>

### token

**Description**: Store token data in a hidden form field (generally used to protect against cross-site forgeries). A token element is automatically added to each Drupal form by [drupal\_prepare\_form](https://api.drupal.org/api/drupal/includes%21form.inc/function/drupal_prepare_form/7.x" \o "Prepares a structured form array.)(), so you don't generally have to add one yourself.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#default\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix),[#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [**#value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#value), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

### value

**Description**: A form value that is internal to the form and never displayed to the screen.

**Properties**: [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [**#value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#value)

**Usage example** ([node.module](https://api.drupal.org/api/drupal/7.x/search/node.module" \o "Multiple implementations exist.)):

<?php  
$form['vid'] = array('#type' => 'value', '#value' => $node->vid);  
?>

Note that as of Drupal 6, you can also simply store arbitrary variables in $form['#foo'] instead, as long as '#foo'does not conflict with any other internal property of the Form API.

### vertical\_tabs

**Description**: Formats all child fieldsets and all non-child fieldsets whose [#group](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#group) is assigned this element's name as vertical tabs.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [**#default\_tab**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_tab), [#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix),[#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Functions that use vertical\_tabs**:  
[block\_admin\_configure](http://api.drupal.org/api/drupal/modules%21block%21block.admin.inc/function/block_admin_configure) in [block.admin.inc](https://api.drupal.org/api/drupal/7.x/search/block.admin.inc)  
[user\_admin\_settings](http://api.drupal.org/api/drupal/modules%21user%21user.admin.inc/function/user_admin_settings) in [user.admin.inc](https://api.drupal.org/api/drupal/modules%21user%21user.admin.inc/7.x)

### weight

**Description**: Format a weight selection menu.

**Properties**: [#access](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#access), [#after\_build](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#after_build), [#attributes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#attributes), [**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), [**#delta**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#delta) (default: 10), [#description](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#description), [#disabled](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#disabled),[#element\_validate](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#element_validate), [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents), [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render), [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix), [#pre\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#pre_render), [#process](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#process), [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [#states](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#states), [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix), [#theme](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme), [#theme\_wrappers](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#theme_wrappers), [**#title**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title), [#title\_display](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#title_display), [#tree](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tree), [**#type**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#type), [#weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weightval)

**Usage example** ([menu.module](https://api.drupal.org/api/drupal/7.x/search/menu.module" \o "Multiple implementations exist.)):

<?php  
$form['weight'] = [array](http://php.net/array)(  
  '#type' => 'weight',   
  '#title' => t('Weight'),   
  '#default\_value' => $edit['weight'],   
  '#delta' => 10,   
  '#description' => t('Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.'),  
);  
?>

## Properties

### #access

**Used by**: All elements and forms

**Description**: Whether the element is accessible or not; when FALSE, the element is not rendered and the user submitted value is not taken into consideration.

**Values**: TRUE or FALSE.

### #action

**Used by**: [form](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#form)

**Description**: The path to which the form will be submitted.

**Values**: An internal path

**Usage example** ([comment.module](https://api.drupal.org/api/drupal/7.x/search/comment.module" \o "Multiple implementations exist.)):

<?php  
$form['#action'] = url('comment/reply/'. $edit['nid']);  
?>

### #after\_build

**Used by**: All elements and forms

**Description**: An array of function names which will be called after the form or element is built.

**Usage example** ([system.admin.inc](https://api.drupal.org/api/drupal/7.x/search/system.admin.inc)):

<?php  
  $form['file\_public\_path'] = [array](http://php.net/array)(  
    '#type' => 'textfield',  
    '#title' => t('Public file system path'),  
    '#default\_value' => variable\_get('file\_public\_path', conf\_path() . '/files'),  
    '#maxlength' => 255,  
    '#description' => t('A local file system path where public files will be stored. This directory must exist and be writable by Drupal. This directory must be relative to the Drupal installation directory and be accessible over the web.'),  
    '#after\_build' => [array](http://php.net/array)('system\_check\_directory'),  
  );  
?>

### [#ajax](http://api.drupal.org/api/drupal/includes--ajax.inc/group/ajax)

**Used by**: [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [checkboxes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkboxes), [image button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#image_button), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [radios](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radios), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit), [tableselect](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tableselect), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#text_format), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield)

An array of elements whose values control the behavior of the element with respect to the [Drupal AJAX framework](http://api.drupal.org/api/drupal/includes--ajax.inc/group/ajax).

[AJAX](http://en.wikipedia.org/wiki/AJAX) (Asynchronous Javascript and XML) is a term used for dynamic communication between the browser and the server, without page reloads. A purist would insist that the Drupal technique is [AHAH](http://en.wikipedia.org/wiki/AHAH) (Asychronous HTML and HTTP) because XML is not used. Until Drupal 7, AJAX was known as AHAH. In a nutshell an AJAX request follows these steps:

1. Drupal builds a form element with a set of #ajax properties. The misc/ajax.js file is included on the page automatically.
2. ajax.js finds all the form elements on the page that have an [#ajax['callback']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_callback) or an [#ajax['path']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_path) set and adds an event handler for the [#ajax['event']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_event) set on that form element.
3. When the [#ajax['event']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_event) occuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuurs on the element (such as 'click'), the AJAX request is made to the Drupal path of the element's [#ajax['path']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_path). If an [#ajax['callback']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_callback) has been specified (the normal case), the Drupal path will be system/ajax.
4. Form information gets processed, and the function specified in [#ajax['callback']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_callback) is called.  [#ajax['callback']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/forms_api_reference.html#ajax_callback) can return either HTML, a renderable array, or an array of [AJAX commands](http://api.drupal.org/api/group/ajax_commands).
5. While the user waits for the callback to execute a throbber or progress bar is shown as determined by [#ajax['progress']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_progress). The result is returned to the original page containing the form element.
6. ajax.js gets the result. If it is HTML and [#ajax['wrapper']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/forms_api_reference.html#ajax_wrapper) is set, the HTML replaces the element specified by [#ajax['wrapper']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/forms_api_reference.html#ajax_wrapper). If it is an array of AJAX commands, the commands are executed. and inserts the returned HTML into the [#ajax['wrapper']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_wrapper).

Example usages of basic AJAX and AJAX commands are provided in the [Examples module](http://drupal.org/project/examples).

### #ajax['callback']

**Description**: Specifies the name of a callback function which will be called during an AJAX call. It can return HTML, a renderable array, or an array of [AJAX Commands](http://api.drupal.org/api/group/ajax_commands). This callback function is given the $form and $form\_stateparameters, allowing it to produce a result, which it returns for rendering.

Note: #ajax['callback'] and #ajax['path'] are mutually exclusive. 'callback' is easier to use and requires less code setup, but may not provide the capabilities required for some functions. When you can, use 'callback'.

**Values**: String containing a function name.

**Usage example** ([poll.module](https://api.drupal.org/api/drupal/7.x/search/poll.module" \o "Multiple implementations exist.)):

<?php  
/\*\*  
 \* Implementation of hook\_form().  
 \*/  
function poll\_form(&$node, $form\_state) {  
  ...  
  $form['choice\_wrapper']['poll\_more'] = array(  
    '#type' => 'submit',  
    '#value' => t('More choices'),  
    '#description' => t("If the amount of boxes above isn't enough, click here to add more choices."),  
    '#weight' => 1,  
    '#submit' => array('poll\_more\_choices\_submit'), // If no javascript action.  
    '#ajax' => array(  
      'callback' => 'poll\_choice\_js',  
      'wrapper' => 'poll-choices',  
      'method' => 'replace',  
      'effect' => 'fade',  
    ),  
  );  
  ...  
}  
  
/\*\*  
 \* Menu callback for AJAX additions. Render the new poll choices.  
 \*/  
function [poll\_choice\_js](https://api.drupal.org/api/drupal/modules%21poll%21poll.module/function/poll_choice_js/7.x)($form, $form\_state) {  
  return $form['choice\_wrapper']['choice'];  
}  
?>

### #ajax['effect']

**Description**: Specifies the effect used when adding the content from an AJAX request.

**Values**: String. Possible values: 'none' (default), 'fade', 'slide'. If the [interface elements library](http://interface.eyecon.ro/) is installed, any effect with the name effectToggle may also be used.

### #ajax['event']

**Description**: When this event occurs to this element, Drupal will perform an HTTP request in the background via Javascript.

**Values**: String. Possible values: Any valid [jQuery event](http://docs.jquery.com/Events), including 'mousedown' (for submit, imagebutton, and button), 'blur' (for textfield and textarea), 'change' (for select). Note that #ajax['event'] does not need to be explicitly specified. Although it can be manually set, usually the default value will be sufficient.

### #ajax['keypress']

**Description**: If set to TRUE, then the element's #ajax['event'] will be triggered if the ENTER key is pressed while the element has focus.

### #ajax['method']

**Description**: Modify the behavior of the returned HTML from an AJAX request when inserting into the [#ajax['wrapper']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_wrapper). If not set, the returned HTML will *replace* the contents of the wrapper element, but it's also possible to use any of the available [jQuery operations for DOM manipulation](http://docs.jquery.com/DOM/Manipulation).

**Values**: String. Possible values: 'replace' (default), 'after', 'append', 'before', 'prepend'.

### #ajax['path']

**Description**: This is the Drupal menu path for a callback function which will generate HTML and return the string of HTML to Drupal. The result will replace the div specified in [#ajax['wrapper']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_wrapper). This property is infrequently used in Drupal 7 and higher, because it is set automatically if using the [#ajax['callback']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_callback) property. When using #ajax['callback'] the path is automatically set to system/ajax, which provides a menu callback that can be used for many situations.

Note: 'path' and 'callback' are mutually exclusive.

**Values**: String containing a Drupal menu path.

### #ajax['prevent']

**Description**: A JavaScript event to prevent when 'event' is triggered. Defaults to 'click' for #ajax on #type 'submit', 'button', and 'image\_button'. Multiple events may be specified separated by spaces. For example, when binding #ajax behaviors to form buttons, pressing the ENTER key within a textfield triggers the 'click' event of the form's first submit button. Triggering Ajax in this situation leads to problems, like breaking autocomplete textfields. Because of that, Ajax behaviors are bound to the 'mousedown' event on form buttons by default. However, binding to 'mousedown' rather than 'click' means that it is possible to trigger a click by pressing the mouse, holding the mouse button down until the Ajax request is complete and the button is re-enabled, and then releasing the mouse button. For this case, 'prevent' can be set to 'click', so an additional event handler is bound to prevent such a click from triggering a non-Ajax form submission. This also prevents a textfield's ENTER press triggering a button's non-Ajax form submission behavior.

**Values**: String. Possible values: One or more [jQuery events](http://docs.jquery.com/Events), with [multiple events](http://api.jquery.com/bind#multiple-events) separated by space characters.

### #ajax['progress']

**Description**: Choose either a throbber or progress bar that is displayed while awaiting a response from the callback, and add an optional message.

**Values**: Array.

Possible keys: 'type', 'message', 'url', 'interval'

Possible values:

* **#ajax['progress']['type']** String. Possible values: 'throbber' (default), 'bar'.
* **#ajax['progress']['message']** String. An optional message to the user; should be wrapped with [t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)().
* **#ajax['progress']['url']** String. The optional callback path to use to determine how full the progress bar is (as defined in progress.js). Only useable when 'type' is 'bar'.
* **#ajax['progress']['interval']** String. The interval to be used in updating the progress bar (as defined in progress.js). Ony used if 'url' is defined and 'type' is 'bar'.

**Usage example** ([file.module](https://api.drupal.org/api/drupal/modules%21file%21file.module/7.x" \o "Defines a \"managed_file\" Form API field and a \"file\" field for Field module.)):

<?php  
  $ajax\_settings['progress']['type'] ? $ajax\_settings['progress']['type'] == 'bar' : 'throbber';  
  $ajax\_settings['progress']['message'] = NULL;  
  $ajax\_settings['effect'] = 'none';  
  $element['remove\_button'] = [array](http://php.net/array)(  
    '#name' => implode('\_', $element['#parents']) . '\_remove\_button',  
    '#type' => 'submit',  
    '#value' => t('Remove'),                                              
    '#validate' => [array](http://php.net/array)(),  
    '#submit' => [array](http://php.net/array)('file\_managed\_file\_submit'),  
    '#limit\_validation\_errors' => [array](http://php.net/array)($element['#parents']),  
    '#ajax' => $ajax\_settings,  
    '#weight' => -5,  
  );  
?>

### #ajax['trigger\_as']

**Description**: For a non-submit, non-button element, allows selection of the submit element which will be activated when the element is triggered. For example, if a select element is ajax-enabled, and it changes, an element identified by #ajax['trigger\_as']['name'] or #ajax['trigger\_as']['value'] will be used as the triggering element, especially for button-level validation.

**Values**: Array.

Possible values

* [array](http://php.net/array)('name' => some\_submit\_or\_button\_name)
* [array](http://php.net/array)('value' => some\_submit\_or\_button\_value)

Note that if #tree == TRUE, #name encodes the parents of the element in it, as when used with [form\_set\_error](https://api.drupal.org/api/drupal/includes%21form.inc/function/form_set_error/7.x" \o "Files an error against a form element.)(), i.e. "level1][level2][element".

### #ajax['wrapper']

**Description**: This property defines the HTML id attribute of an element on the page which will be replaced by the html returned by the [#ajax['path']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_path) or [#ajax['callback']](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#ajax_callback) function. Usually, a div element is used as the wrapper, as it provides the most flexibility for placement of elements before, after, or inside of its HTML tags. This property is required for using AJAX requests in on a form element.  Note that the entire element with this ID is replaced, not just the contents of the element.

**Values**: String containg a valid id attribute of an HTML element on the same page. This *must not contain* the '#' character that a selector would have.

**Usage example** ([upload.module](http://api.drupal.org/api/file/modules/upload/upload.module)):

    $form['new']['attach'] = [array](http://php.net/array)(  
      '#type' => 'submit',  
      '#value' => t('Attach'),  
      '#name' => 'attach',  
      '#ajax' => [array](http://php.net/array)(  
        'path' => 'upload/js',  
        'wrapper' => 'attach-wrapper',  
        'progress' => [array](http://php.net/array)('type' => 'bar', 'message' => t('Please wait...')),  
      ),  
      '#submit' => [array](http://php.net/array)('node\_form\_submit\_build\_node'),  
    );

### #array\_parents

**Used by**: All form elements

**Description**: (read-only, and mostly **internal**). The array of names of the element's parents (including itself) in the form. This will always match the structure of $form. It is different from [#parents](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#parents) in that #parents lists only the structure used in $form\_state['values'], which is flat unless #tree is set to TRUE. Please see the [complete comparison between #parents and #array\_parents](http://drupal.org/node/279246#comment-2292322).

### #attached

**Used by**: All form elements

**Description**: Allows loading of CSS, Javascript, libraries, or custom types when the form is built.

**Values**: A keyed array of type => value pairs, where the type (most often 'css', 'js', and 'library') determines the loading technique, and the value provides the options presented to the loader function. See the example below and the loader functions [drupal\_add\_css](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_add_css/7.x)(), [drupal\_add\_js](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_add_js/7.x)(), [drupal\_add\_library](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_add_library/7.x)(), etc.

See [drupal\_process\_attached](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_process_attached/7.x" \o "Adds attachments to a render() structure.)() for additional information.

**Usage example** ([AJAX Example](http://api.drupal.org/api/function/ajax_example_dependent_dropdown_degrades)):

$form['#attached']['css'] = [array](http://php.net/array)(  
[drupal\_get\_path](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_get_path/7.x)('module', 'ajax\_example') . '/ajax\_example.css',  
);  
$form['#attached']['js'] = [array](http://php.net/array)(  
[drupal\_get\_path](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_get_path/7.x)('module', 'ajax\_example') . '/ajax\_example.js',  
);

The above javascript #attach could also be written as:

$form['#attached']['js'] = [array](http://php.net/array)(  
  drupal\_get\_path('module', 'ajax\_example') . '/ajax\_example.js' => [array](http://php.net/array)(  
  'type' => 'file',  
  ),  
);

and inline javascript can be loaded with the javascript as the key of the array, as described in [drupal\_add\_js](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_add_js/7.x" \o "Adds a JavaScript file, setting, or inline code to the page.)().

**Adding settings**

You add a javascript setting by using the 'data' key (as php keys can't be arrays):

$settings = [array](http://php.net/array)('id' => 'mymodule-element-1');  
$form['#attached']['js'][] = [array](http://php.net/array)(  
  'data' => [array](http://php.net/array)('mymodule' => $settings),  
  'type' => 'setting',  
);

**Usage example** ([contact.pages.inc](https://api.drupal.org/api/drupal/7.x/search/contact.pages.inc)):

<?php  
if (!$user->uid) {  
$form['#attached']['library'][] = [array](http://php.net/array)('system', 'jquery.cookie');  
$form['#attributes']['class'][] = 'user-info-from-cookie';  
}  
?>

### #attributes

**Used by**: [actions](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#actions), [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [checkboxes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkboxes), [container](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#container), [date](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#date), [fieldset](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#fieldset), [file](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#file), [form](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#form), [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#image_button), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password),[radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [radios](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radios), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit), [tableselect](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tableselect), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#text_format), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield), [weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weight)

**Description**: Additional HTML attributes, such as 'class' can be set using this mechanism.

**Values**: Any HTML attribute not covered by other properties, e.g. **class** (for control types), **enctype** (for forms).

**Usage example** ([search.module](https://api.drupal.org/api/drupal/7.x/search/search.module" \o "Multiple implementations exist.)):

**Note:** If you are altering an existing form via [hook\_form\_alter](https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook_form_alter/7.x" \o "Perform alterations before a form is rendered.)() or a similar means, be careful with this property! You will probably want to add to the existing array rather than writing over it, so don't follow this usage example exactly.

<?php  
$form['#attributes'] = [array](http://php.net/array)('class' => [array](http://php.net/array)('search-form'));  
?>

### #autocomplete\_path

**Used by**: [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "textfield)

**Description**: The path the AJAX autocomplete script uses as the source for autocompletion.

**Usage example** ([node.module](https://api.drupal.org/api/drupal/7.x/search/node.module" \o "Multiple implementations exist.)):

<?php  
$form['author']['name'] = [array](http://php.net/array)(  
  '#type' => 'textfield',   
  '#title' => t('Authored by'),   
  '#maxlength' => 60,   
  '#autocomplete\_path' => 'user/autocomplete',   
  '#default\_value' => $node->name,   
  '#weight' => -1,  
);  
?>

### #built

**Used by**: [form](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#form)

**Description**: Used to ascertain whether or not a form element has been built yet.

**Values**: TRUE or FALSE

**Usage example**: INTERNAL. See the [\_form\_builder](http://api.drupal.org/api/function/_form_builder) function in [form.inc](https://api.drupal.org/api/drupal/7.x/search/form.inc).

### #button\_type

**Used by**: [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "image_button), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit)

**Description**: Adds a CSS class to the button, in the form form-[button\_type\_value]. Note that this does NOT set the HTML attribute type of the button.

**Values**: String

**Usage example**: ([system.module](https://api.drupal.org/api/drupal/7.x/search/system.module" \o "Multiple implementations exist.)):

$types['submit'] = [array](http://php.net/array)(  
  '#input' => TRUE,  
  '#name' => 'op',  
  '#button\_type' => 'submit',  
  '#executes\_submit\_callback' => TRUE,  
  '#limit\_validation\_errors' => FALSE,  
  '#process' => [array](http://php.net/array)('ajax\_process\_form'),  
  '#theme\_wrappers' => [array](http://php.net/array)('button'),  
);

### #children

**Used by**: All elements.

**Description**: Internal. The rendered child elements of an element. Automatically assigned by [drupal\_render](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_render/7.x" \o "Renders HTML given a structured array tree.)(), unless already set. In rare cases, you may use #children to override the output of rendered child elements, for example in a #theme\_wrappers or #post\_render callback. Normally, you should not have to use #children, use regular child elements instead. (See [actions](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#actions) for an example.)

**Values**: String representing the rendered child elements of an element.

**Usage example** [theme\_container](https://api.drupal.org/api/drupal/includes%21form.inc/function/theme_container/7.x" \o "Returns HTML to wrap child elements in a container.)()):   
Example in a #theme\_wrappers callback.

<?php  
function theme\_container($variables) {  
  $element = $variables['element'];  
  
  // ...  
  
  return '<div' . drupal\_attributes($element['#attributes']) . '>' . $element['#children'] . '</div>';  
}  
?>

### #collapsed

**Used by**: [fieldset](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "fieldset)

**Description**: Indicates whether or not the fieldset is collapsed by default. See [#collapsible](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#collapsible) property.

**Values**: TRUE or FALSE

**Usage example** ([block.module](https://api.drupal.org/api/drupal/7.x/search/block.module" \o "Multiple implementations exist.)) :

<?php  
$form['block'] = [array](http://php.net/array)(  
  '#type' => 'fieldset',   
  '#title' => t('Block configuration'),   
  '#weight' => 3,   
  '#collapsible' => TRUE,   
  '#collapsed' => FALSE,   
  '#tree' => TRUE,  
);  
?>

### #collapsible

**Used by**: [fieldset](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "fieldset)

**Description**: Indicates whether or not the fieldset can be collapsed with JavaScript. See [#collapsed](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#collapsed) property.

**Values**: TRUE or FALSE

**Usage example** ([block.module](https://api.drupal.org/api/drupal/7.x/search/block.module" \o "Multiple implementations exist.)):

<?php  
$form['block'] = [array](http://php.net/array)(  
  '#type' => 'fieldset',   
  '#title' => t('Block configuration'),   
  '#weight' => 3,   
  '#collapsible' => TRUE,   
  '#collapsed' => FALSE,   
  '#tree' => TRUE,  
);  
?>

### #cols

**Used by**: [text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "text_format), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "textarea)

**Description**: How many columns wide the textarea should be (see also [#rows](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#rows))

**Values**: A positive number

**Usage example** ([aggregator.module](https://api.drupal.org/api/drupal/modules%21aggregator%21aggregator.module/7.x" \o "Used to aggregate syndicated content (RSS, RDF, and Atom).)):

<?php  
$form['description'] = [array](http://php.net/array)(  
  '#type' => 'textarea',   
  '#title' => t('Description'),   
  '#default\_value' => $edit['description'],   
  '#cols' => 60,   
  '#rows' => 5,  
);  
?>

### #default\_tab

**Used by**: [vertical\_tabs](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "vertical_tabs)

**Description**: The default open tab.

**Values**: String containing the HTML ID of the fieldset element for the default tab. The default value is an empty string, which results in the first tab having the focus. The fieldset's HTML ID typically takes the form of "edit-ARRAY\_KEY".

**Usage example**:

<?php  
$form['vertical\_tabs'] = [array](http://php.net/array)(  
'#type' => 'vertical\_tabs',  
'#default\_tab' => 'edit-tab2',  
);  
$form['tab1'] = [array](http://php.net/array)(  
'#type' => 'fieldset',  
'#title' => t('Tab 1'),  
'#collapsible' => TRUE,  
'#group' => 'vertical\_tabs',  
);  
$form['tab1']['field1'] = [array](http://php.net/array)(  
'#title' => t('Field 1'),  
'#type' => 'textfield',  
);  
$form['tab2'] = [array](http://php.net/array)(  
'#type' => 'fieldset',  
'#title' => t('Tab 2'),  
'#collapsible' => TRUE,  
'#group' => 'vertical\_tabs',  
);  
$form['tab2']['field2'] = [array](http://php.net/array)(  
'#title' => t('Field 2'),  
'#type' => 'textfield',  
);  
?>

### #default\_value

**Used by**: [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [checkboxes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkboxes), [date](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#date), [hidden](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#hidden), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [radios](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radios), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [tableselect](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tableselect), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#text_format), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield), [token](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#token), [weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weight)

**Description**: The value of the form element that will be displayed or selected initially if the form has not been submitted yet. **Should NOT be confused with** [**#value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#value), which is a hard-coded value the user cannot change!

**Values**: Mixed

**Usage example** ([forum.module](https://api.drupal.org/api/drupal/7.x/search/forum.module" \o "Multiple implementations exist.)):

<?php  
$form['body'] = [array](http://php.net/array)(  
  '#type' => 'textarea',   
  '#title' => t('Body'),   
  '#default\_value' => $node->body,   
  '#required' => TRUE,  
);  
?>

### #delta

**Used by**: [weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weight)

**Description**: Number of weights to have selectable. For example, with $delta => 10, the weight selection box would display numbers from -10 to 10.

**Values**: A positive number

**Usage example** ([menu.module](https://api.drupal.org/api/drupal/7.x/search/menu.module" \o "Multiple implementations exist.)):

<?php  
$form['weight'] = [array](http://php.net/array)(  
  '#type' => 'weight',   
  '#title' => t('Weight'),   
  '#default\_value' => $edit['weight'],   
  '#delta' => 10,   
  '#description' => t('Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.'),  
);  
?>

### #description

**Used by**: [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [checkboxes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkboxes), [date](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#date), [fieldset](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#fieldset), [file](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#file), [item](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#item), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [password\_confirm](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password_confirm), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [radios](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radios), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select),[text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#text_format), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield), [weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weight)

**Description**: The description of the form element. Make sure to enclose inside the [t](http://api.drupal.org/api/function/t)() function so this property can be translated.

**Values**: Mixed

**Usage example** ([menu.module](https://api.drupal.org/api/drupal/7.x/search/menu.module" \o "Multiple implementations exist.)):

<?php  
$form['weight'] = [array](http://php.net/array)(  
  '#type' => 'weight',   
  '#title' => t('Weight'),   
  '#default\_value' => $edit['weight'],   
  '#delta' => 10,   
  '#description' => t('Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.'),  
);  
?>

### #disabled

**Used by**: [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [checkboxes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkboxes), [date](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#date), [file](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#file), [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#image_button), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [password\_confirm](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password_confirm), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [radios](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radios), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit), [text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#text_format), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield), [weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weight)

**Description**: Disables (greys out) a form input element. Setting #disabled to TRUE results in user input being ignored, regardless of how the element is themed or whether JavaScript is used to change the control's attributes.

**Values**: TRUE or FALSE

**Usage example** ([system.module](https://api.drupal.org/api/drupal/7.x/search/system.module" \o "Multiple implementations exist.)):

<?php  
      if ([isset](http://php.net/isset" \o )($disabled[$name])) {  
        $form['theme\_settings'][$name]['#disabled'] = TRUE;  
      }  
?>

### #element\_validate

**Used by**: Any element.

**Description**: A list of custom validation functions. The validation functions must use [form\_error](https://api.drupal.org/api/drupal/includes%21form.inc/function/form_error/7.x" \o "Flags an element as having an error.)() or [form\_set\_error](https://api.drupal.org/api/drupal/includes%21form.inc/function/form_set_error/7.x" \o "Files an error against a form element.)() to flag the element as having an error if the validation fails.

**Values**: An array of function names to be called to validate this element (and/or its children).

A validation function for an element takes $element, $form\_state, and $form as parameters, and has the form:

function myelement\_validate($element, &$form\_state, $form) {  
   if ([empty](http://php.net/empty)($element['#value'])) {  
     form\_error($element, t('This field is required.'));  
   }  
}

**Usage example** ([file.field.inc](https://api.drupal.org/api/drupal/7.x/search/file.field.inc)):

**Note:** If you are altering an existing form element via [hook\_form\_alter](https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook_form_alter/7.x" \o "Perform alterations before a form is rendered.)() or a similar means, be careful with this property! You will probably want to add to the existing array rather than writing over it, so in that case don't follow this usage example exactly.

You'll notice that the validation function below includes only the first two arguments. $form is the same as $form\_state['complete form'], so the third parameter is in fact contained in the second parameter.

<?php  
$form['max\_filesize'] = [array](http://php.net/array)(  
    '#type' => 'textfield',  
    '#title' => t('Maximum upload size'),  
    '#default\_value' => $settings['max\_filesize'],  
    '#description' => t('Enter a value like "512" (bytes), "80 KB" (kilobytes) or "50 MB" (megabytes) in order to restrict the allowed file size. If left empty the file sizes will be limited only by PHP\'s maximum post and file upload sizes (current limit <strong>%limit</strong>).', [array](http://php.net/array)('%limit' => format\_size(file\_upload\_max\_size()))),  
    '#size' => 10,  
    '#element\_validate' => [array](http://php.net/array)('\_file\_generic\_settings\_max\_filesize'),  
    '#weight' => 5,  
  );  
...  
function \_file\_generic\_settings\_max\_filesize($element, &$form\_state) {  
  if (!empty($element['#value']) && !is\_numeric(parse\_size($element['#value']))) {  
    form\_error($element, t('The "!name" option must contain a valid value. You may either leave the text field empty or enter a string like "512" (bytes), "80 KB" (kilobytes) or "50 MB" (megabytes).', [array](http://php.net/array)('!name' => t($element['title']))));  
  }  
}   
?>

### #empty

**Used by**: [tableselect](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "tableselect)

**Description**: Text to display if the #options property is empty.

**Values**: Text, enclosed in the [t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)() translation function.

### #empty\_option

**Used by**: [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select)

**Description**: The label to show for the initial option denoting no selection in a select element. By default, the label is automatically set to "- Select -" for a required field and "- None -" for an optional field.

**Values**: Text, enclosed in the [t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)() translation function.

**Note:** Use of this property plays a part in several possible states for a select field:

* If [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required) is TRUE and there is no [#default\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), an empty option is added to the select control to force the user to make an active choice.
* If [#empty\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#empty_value) or #empty\_option is set and [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required) is FALSE (default), an empty option is added to the select control, allowing the user to choose nothing.
* If none of [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), [#empty\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#empty_value), #empty\_option, and [#default\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value) are set, then no empty option is added to the select control. This leaves the control in a slightly illogical state, since all user agents automatically preselect the first available option.There's no way for the user to select nothing, and the user is also not forced to make an active decision.

### #empty\_value

**Used by**: [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select)

**Description**: The value for the initial option denoting no selection in a select element, which is used to determine whether the user submitted a value or not.

**Values**: Defaults to empty string. Can be anything except NULL.

**Note:** Use of this property plays a part in several possible states for a select field:

* If [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required) is TRUE and there is no [#default\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), an empty option is added to the select control to force the user to make an active choice.
* If #empty\_value or [#empty\_option](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#empty_option) is set and [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required) is FALSE (default), an empty option is added to the select control, allowing the user to choose nothing.
* If none of [#required](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#required), #empty\_value, [#empty\_option](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#empty_option), and [#default\_value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value) are set, then no empty option is added to the select control. This leaves the control in a slightly illogical state, since all user agents automatically preselect the first available option. There's no way for the user to select nothing, and the user is also not forced to make an active decision.

### #error

INTERNAL. Indicates whether or not a form element has been flagged as having an error.

### #executes\_submit\_callback

**Used by**: [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "image_button), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit)

**Description**: Indicates whether or not the submit handler should be run when the form is submitted using this button element.

**Values**: TRUE or FALSE

### #field\_prefix

**Used by**: [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [machine\_name](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#machine_name), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [password\_confirm](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password_confirm), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield)

**Description**: Text or code that is placed directly in front of the textfield. This can be used to prefix a textfield with a constant string.

**Values**: Mixed

**Usage example** ([system.module](https://api.drupal.org/api/drupal/7.x/search/system.module" \o "Multiple implementations exist.)):

<?php  
$form['site\_403'] = array(  
  '#type' => 'textfield',  
  '#title' => t('Default 403 (access denied) page'),  
  '#default\_value' => variable\_get('site\_403', ''),  
  '#size' => 40,  
  '#description' => t('This page is displayed when the requested document is denied to the current user. If unsure, specify nothing.'),  
  '#field\_prefix' => url(NULL, NULL, NULL, TRUE) . (variable\_get('clean\_url', 0) ? '' : '?q=')  
);  
?>

### #field\_suffix

**Used by**: [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [machine\_name](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#machine_name), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [password\_confirm](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password_confirm), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield)

**Description**: Text or code that is placed directly after a textfield. This can be used to add a unit to a textfield.

**Values**: Mixed

**Usage example** ([system.module](https://api.drupal.org/api/drupal/7.x/search/system.module" \o "Multiple implementations exist.)):

<?php  
$form['settings\_general']['upload\_usersize\_default'] = array(  
  '#type' => 'textfield',  
  '#title' => t('Default total file size per user'),  
  '#default\_value' => $upload\_usersize\_default,  
  '#size' => 5,  
  '#maxlength' => 5,  
  '#description' => t('The default maximum size of all files a user can have on the site.'),  
  '#field\_suffix' => t('MB')  
);  
?>

### #group

**Used by**: [fieldset](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "fieldset)

**Description**: Specifies a group that can be used to group fieldsets into vertical tabs.

**Values**: A string where the value is used for the id of a form element with the #type of vertical\_tabs.

**Usage example** ([user.admin.inc](https://api.drupal.org/api/drupal/modules%21user%21user.admin.inc/7.x)):

<?php  
$form['email'] = array(  
  '#type' => 'vertical\_tabs',  
);  
...  
$form['email\_admin\_created'] = array(  
  '#type' => 'fieldset',  
  '#title' => t('Welcome (new user created by administrator)'),  
  '#collapsible' => TRUE,  
  '#collapsed' => ([variable\_get](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/variable_get/7.x" \o "Returns a persistent variable.)('user\_register', 1) != 0),  
  '#description' => t('Edit the welcome e-mail messages sent to new member accounts created by an administrator.'),  
  '#group' => 'email'  
);  
?>

### #header

**Used by**: [tableselect](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "tableselect)

**Description**: Column headers.

**Values**: Associative array where the keys are the field names to use for each column, and the values are the translated text to display for each column header.

### #id

**Description**: Internal. Used to populate form elements' id property. In rare cases, you can set this value yourself on a form element, to override the default setting.

**Value**: HTML id string.

### #input

INTERNAL. Indicates whether or not input is possible for this form element.

### #js\_select

**Used by**: [tableselect](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "tableselect)

**Description**: Whether to include a select all checkbox.

**Values**: TRUE to add a select all box to each row of the table, FALSE to omit the select all box. Only applies if #multiple is also TRUE.

### #limit\_validation\_errors

**Used by**: [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "image_button), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit)

**Description**: Provides an array of sections which are parts of $form\_state['values'] which should be validated, implying that sections which are not listed should not be validated. This is normally used in multistep forms in the case of a "back" button, for example, where '#limit\_validation\_errors' => [array](http://php.net/array)() would mean not to validate anything as form values on the current page are to be discarded anyway. #limit\_validation\_errors does not have any effect if #submit is not set. More discussion is in the [form\_set\_error](https://api.drupal.org/api/drupal/includes%21form.inc/function/form_set_error/7.x" \o "Files an error against a form element.)() documentation.

**Values**: array indicating sections of the $form\_state['values'] which should be validated.

### #markup

**Used by**: [item](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#item), [markup](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "markup)

**Description**: Used to set HTML that will be output on the form.

**Values**: Text (valid HTML)

**Usage example** ([contact.module](https://api.drupal.org/api/drupal/modules%21contact%21contact.module/7.x" \o "Enables the use of personal and site-wide contact forms.)):

<?php  
  $form['from'] = [array](http://php.net/array)(  
    '#type' => 'item',  
    '#title' => t('From'),  
    '#markup' => check\_plain($user->name) . ' &lt;' . check\_plain($user->mail) . '&gt;',  
  );  
?>

### #maxlength

**Used by**: [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "textfield)

**Description**: The maximum amount of characters to accept as input.

**Values**: A positive number.

**Usage example** ([forum.module](https://api.drupal.org/api/drupal/7.x/search/forum.module" \o "Multiple implementations exist.)):

<?php  
$form['title'] = [array](http://php.net/array)(  
  '#type' => 'textfield',   
  '#title' => t('Subject'),   
  '#default\_value' => $node->title,   
  '#size' => 60,   
  '#maxlength' => 128,   
  '#required' => TRUE,  
);  
?>

### #method

**Used by**: [form](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#form)

**Description**: The HTTP method with which the form will be submitted.

**Values**: get or post. Default is post.

**Usage example** ([node.module](https://api.drupal.org/api/drupal/7.x/search/node.module" \o "Multiple implementations exist.)):

<?php  
$form['#method'] = 'post';  
?>

### #multiple

**Used by**: [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [tableselect](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "tableselect)

**Description**: Indicates whether the user may select more than one item.

**Values**: TRUE or FALSE

**Usage example** ([taxonomy.module](https://api.drupal.org/api/drupal/7.x/search/taxonomy.module" \o "Multiple implementations exist.)):

<?php  
return [array](http://php.net/array)(  
  '#type' => 'select',   
  '#title' => $title,   
  '#default\_value' => $value,   
  '#options' => $options,   
  '#description' => $description,   
  '#multiple' => $multiple,   
  '#size' => $multiple ? min(12, count($options)) : 0,  
  '#weight' => -15,  
);  
?>

### #name

**Used by**: [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit), [file](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#file)

**Description**: The name attribute of the corresponding HTML element. This can be used in element types such as [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit), [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button) and [file](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#file) to override the default setting.

**Values**: String.

### #options

**Used by**: [checkboxes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkboxes), [radios](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radios), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [tableselect](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "tableselect)

**Description**: Selectable options for a form element that allows multiple choices.

**Values**: An array in the form of [array](http://php.net/array)([t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Display value 1'), [t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Display value 2')) or [array](http://php.net/array)('return\_value1' => [t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Display Value 1'), 'return\_value2' => [t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Display Value 2')) if specific return values are required. If returned values are identical to displayed values, you can use [drupal\_map\_assoc](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_map_assoc/7.x" \o "Forms an associative array from a linear array.)() as in the example below. The #options property needs special handling for checkboxes, and works completely differently for tableselect. See the respective element descriptions for details.

**Usage example** ([form\_example\_states.inc](http://api.drupal.org/api/examples/form_example--form_example_states.inc)):

<?php  
  $form['undergraduate']['school\_country'] = [array](http://php.net/array)(  
    '#type' => 'select',  
    '#options' => drupal\_map\_assoc([array](http://php.net/array)(t('UK'), t('Other'))),  
    '#title' => t('In what country is your college or university located?'),  
  );  
?>

### #parents

**Used by**: All

**Description**: Identifies parent form elements. See [#tree and #parents](http://drupal.org/node/48643) in the handbook and the [complete comparison between #parents and #array\_parents](http://drupal.org/node/279246#comment-2292322).

**Values**: An array of element names.

**Usage example** ([comment.module](https://api.drupal.org/api/drupal/7.x/search/comment.module" \o "Multiple implementations exist.)):

<?php  
$form['admin']['status'] = [array](http://php.net/array)(  
  '#type' => 'radios',   
  '#parents' => [array](http://php.net/array)('status'),   
  '#title' => t('Status'),   
  '#default\_value' => $status,   
  '#options' => [array](http://php.net/array)(t('Published'), t('Not published')),   
  '#weight' => -1,  
);  
?>

### #post\_render

**Used by**: All elements and forms

**Description**: Function(s) to call **after** rendering in [drupal\_render](http://api.drupal.org/api/function/drupal_render/)() has occured. The named function is called with two arguments, the rendered element and its children. It returns the (potentially) altered) element content.

**Values**: An array of function names to call.

**Usage example**:

**Note:** If you are altering an existing form via [hook\_form\_alter](https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook_form_alter/7.x" \o "Perform alterations before a form is rendered.)() or a similar means, be careful with this property! You will probably want to add to the existing array rather than writing over it, so don't follow this usage example exactly.

<?php  
$form['admin']['#post\_render'] = [array](http://php.net/array)('admin\_form\_html\_cleanup');  
?>

This example would call admin\_form\_html\_cleanup($content, $element) after $element has been rendered to $content via [drupal\_render](http://api.drupal.org/api/function/drupal_render/)(). $content may then be modified and must be returned.

### #prefix

**Used by**: All elements and forms.

**Description**: Text or markup to include before the form element. Also see [#suffix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#suffix).

**Values**: Mixed

**Usage example** ([poll.module](https://api.drupal.org/api/drupal/7.x/search/poll.module" \o "Multiple implementations exist.)):

<?php  
$form['choice'] = [array](http://php.net/array)(  
  '#type' => 'fieldset',   
  '#title' => t('Choices'),   
  '#prefix' => '<div class="poll-form">',   
  '#suffix' => '</div>',   
  '#tree' => TRUE,  
);  
?>

### #pre\_render

**Used by**: All elements and forms.

**Description**: Function(s) to call **before** rendering in [drupal\_render](http://api.drupal.org/api/function/drupal_render/)() has occured. The function(s) provided in #pre\_render receive the element as an argument and must return the altered element.

**Values**: An array of function names to call.

**Usage example ([filter.module](https://api.drupal.org/api/drupal/modules%21filter%21filter.module/7.x" \o "Framework for handling the filtering of content.))**:

**Note:** If you are altering an existing form via [hook\_form\_alter](https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook_form_alter/7.x" \o "Perform alterations before a form is rendered.)() or a similar means, be careful with this property! As demonstrated here, you will probably want to add to the existing array rather than writing over it.

// Prepend #pre\_render callback to replace field value with user notice  
// prior to rendering.  
if (!isset($element['value']['#pre\_render'])) {  
  $element['value']['#pre\_render'] = [array](http://php.net/array)();  
}  
[array\_unshift](http://php.net/array_unshift)($element['value']['#pre\_render'], 'filter\_form\_access\_denied');  
  
...  
function [filter\_form\_access\_denied](https://api.drupal.org/api/drupal/modules%21filter%21filter.module/function/filter_form_access_denied/7.x)($element) {  
  $element['#value'] = [t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('This field has been disabled because you do not have sufficient permissions to edit it.');  
  return $element;  
}

This example would call [filter\_form\_access\_denied](https://api.drupal.org/api/drupal/modules%21filter%21filter.module/function/filter_form_access_denied/7.x" \o "Render API callback: Hides the field value of 'text_format' elements.)($element) before the $element has been rendered via [drupal\_render](http://api.drupal.org/api/function/drupal_render/)(). $element may then be modified and is returned for rendering.

### #printed

INTERNAL. Used to determine whether or not a form element has been printed yet.

### #process

**Used by**: All

**Description**: An array of functions that are called when an element is processed. Using this callback, modules can "register" further actions. For example, the "radios" form type is expanded to multiple radio buttons using a processing function.

**Values**: Array of function names (strings)

**Usage example** ([system.module](https://api.drupal.org/api/drupal/7.x/search/system.module" \o "Multiple implementations exist.)) in [system\_element\_info](https://api.drupal.org/api/drupal/modules%21system%21system.module/function/system_element_info/7.x" \o "Implements hook_element_info().)():

**Note:** If you are altering an existing form via [hook\_form\_alter](https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook_form_alter/7.x" \o "Perform alterations before a form is rendered.)() or a similar means, be careful with this property! You will probably want to add to the existing array rather than writing over it, so don't follow this usage example exactly.

<?php  
$types['radios'] = [array](http://php.net/array)(  
'#input' => TRUE,   
'#process' => [array](http://php.net/array)('form\_process\_radios'),   
'#theme\_wrappers' => [array](http://php.net/array)('radios'),   
'#pre\_render' => [array](http://php.net/array)('form\_pre\_render\_conditional\_form\_element'),  
);  
?>

In this example, the function form\_process\_radios is called; the entire radios element is passed in as the argument. This function expands a radios element into individual radio buttons.

### #processed

INTERNAL. Used to ascertain whether or not a form element has been processed (ie: expanded to multiple elements).

### #required

**Used by**: [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [checkboxes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkboxes), [date](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#date), [file](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#file), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [password\_confirm](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password_confirm), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [radios](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radios), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#text_format), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield), [weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weight)

**Description**: Indicates whether or not the element is required. This automatically validates for empty fields, and flags inputs as required. File fields are **NOT** allowed to be required.

**Values**: TRUE or FALSE

**Usage example** ([forum.module](https://api.drupal.org/api/drupal/7.x/search/forum.module" \o "Multiple implementations exist.)):

<?php  
$form['title'] = [array](http://php.net/array)(  
  '#type' => 'textfield',   
  '#title' => t('Subject'),   
  '#default\_value' => $node->title,   
  '#size' => 60,   
  '#maxlength' => 128,   
  '#required' => TRUE,  
);  
?>

### #resizable

**Used by**: [text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "text_format), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "textarea)

**Description**: Whether users should be allowed to resize the text area

**Values**: TRUE or FALSE

### #return\_value

**Used by**: [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "image_button), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio)

**Description**: Value element should return when selected

**Values**: Mixed

**Usage example** ([poll.module](https://api.drupal.org/api/drupal/7.x/search/poll.module" \o "Multiple implementations exist.)):

<?php  
$form['morechoices'] = [array](http://php.net/array)(  
  '#type' => 'checkbox',   
  '#title' => t('Need more choices'),   
  '#return\_value' => 1,   
  '#default\_value' => 0,   
  '#description' => t("If the amount of boxes above isn't enough, check this box and click the Preview button below to add some more."),  
);  
?>

### #rows

**Used by**: [text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "text_format), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "textarea)

**Description**: How many rows high the textarea should be (see also [#cols](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#cols))

**Values**: A positive number

**Usage example** ([aggregator.module](https://api.drupal.org/api/drupal/modules%21aggregator%21aggregator.module/7.x" \o "Used to aggregate syndicated content (RSS, RDF, and Atom).)):

<?php  
$form['description'] = [array](http://php.net/array)(  
  '#type' => 'textarea',   
  '#title' => t('Description'),   
  '#default\_value' => $edit['description'],   
  '#cols' => 60,   
  '#rows' => 5,  
);  
?>

### #size

**Used by**: [file](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#file), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [password\_confirm](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "password_confirm), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "textfield)

**Description**: Width of the textfield (in characters) or size of multiselect box (in lines).

**Values**: A positive number.

**Usage example** ([comment.module](https://api.drupal.org/api/drupal/7.x/search/comment.module" \o "Multiple implementations exist.)):

<?php  
        $form['admin']['homepage'] = [array](http://php.net/array)(  
          '#type'=> 'textfield',  
          '#title'=> t('Homepage'),  
          '#maxlength'=> 255,  
          '#size'=> 30,  
          '#default\_value'=> $edit['homepage'],  
        );  
?>

### #src

**Used by**: [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "image_button)

**Description**:  The URL of the image of the button.

**Values**: A URL.

### #states

**Used by**: any element

**Description**:  Adds JavaScript to the element to allow it to have different active states.

**Values**: A structured array describing the different JavaScript states that can be applied to the element when certain conditions are met. Please read the extensive description of this feature at [drupal\_process\_states](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_process_states/7.x" \o "Adds JavaScript to change the state of an element based on another element.)().

**Usage example**

$form['email\_canceled']['settings'] = [array](http://php.net/array)(  
    '#type' => 'container',  
    '#states' => [array](http://php.net/array)(  
      // Hide the settings when the cancel notify checkbox is disabled.  
      'invisible' => [array](http://php.net/array)(  
       ':input[name="email\_canceled\_toggle"]' => [array](http://php.net/array)('checked' => FALSE),  
      ),  
    ),  
  );

### #submit

**Used by**: [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "image_button), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit), [form](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#form)

**Description**:  A list of custom submit functions that will be called when the element is submitted. This is usually to add additional submit functions to a form, or to use an alternate function rather than the default form submission function which is the form ID with \_submit appended to it.

**Values**: An array of function names. Each such function will take $form and $form\_state as parameters.

**Usage example** ([node\_form](https://api.drupal.org/api/drupal/modules%21node%21node.pages.inc/function/node_form/7.x" \o "Form constructor for the node add/edit form.)()):

if (!empty($node->nid) && [node\_access](https://api.drupal.org/api/drupal/modules%21node%21node.module/function/node_access/7.x)('delete', $node)) {  
 $form['actions']['delete'] = [array](http://php.net/array)(  
  '#type' => 'submit',  
  '#value' => [t](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Delete'),  
  '#weight' => 15,  
  '#submit' => [array](http://php.net/array)('node\_form\_delete\_submit'),  
 );

### #suffix

**Used by**: All elements and forms.

**Description**: Text or markup to include after the form element. Also see [#prefix](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#prefix).

**Values**: Mixed

**Usage example** ([poll.module](https://api.drupal.org/api/drupal/7.x/search/poll.module" \o "Multiple implementations exist.)):

<?php  
$form['choice'] = [array](http://php.net/array)(  
  '#type' => 'fieldset',   
  '#title' => t('Choices'),   
  '#prefix' => '<div class="poll-form">',   
  '#suffix' => '</div>',   
  '#tree' => TRUE,  
);  
?>

### #theme

**Used by**: All elements and forms.

**Description**: Theme function to call for element. It must render this element and all of its child elements.

**Values**: The name of a theme function, without the initial 'theme\_'.

**Usage example** ([node.module](https://api.drupal.org/api/drupal/7.x/search/node.module" \o "Multiple implementations exist.)):

<?php  
$data['actions']['output'][] = [array](http://php.net/array)(  
        '#theme' => 'menu\_local\_action',  
        '#link' => $item,  
      );  
?>

### #theme\_wrappers

**Used by**: All elements and forms.

**Description**: Theme function to call for element, after the element and children are rendered, but before the [#post\_render](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#post_render) functions are called.

**Values**: The name of a theme function, without the initial 'theme\_'.

**Usage example** ([system.module](https://api.drupal.org/api/drupal/7.x/search/system.module" \o "Multiple implementations exist.) from definition of default properties for a form):

<?php  
$type['form'] = [array](http://php.net/array)('#theme\_wrappers' => [array](http://php.net/array)('form'));   
?>

### #title

**Used by**: [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [checkboxes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkboxes), [date](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#date), [fieldset](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#fieldset), [file](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#file), [item](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#item), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [password\_confirm](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password_confirm), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [radios](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radios), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select),[text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#text_format), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield), [weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weight)

**Description**: The title of the form element. Make sure to enclose inside the [t](http://api.drupal.org/api/function/t)() function so this property can be translated.

**Values**: Mixed

**Usage example** ([aggregator.module](https://api.drupal.org/api/drupal/modules%21aggregator%21aggregator.module/7.x" \o "Used to aggregate syndicated content (RSS, RDF, and Atom).)):

<?php  
$form['description'] = [array](http://php.net/array)(  
  '#type' => 'textarea',   
  '#title' => t('Description'),   
  '#default\_value' => $edit['description'],   
  '#cols' => 60,   
  '#rows' => 5,  
);  
?>

### #title\_display

**Used by**: [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [checkboxes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkboxes), [date](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#date), [fieldset](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#fieldset), [file](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#file), [item](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#item), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [password\_confirm](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password_confirm), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [radios](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radios), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [text\_format](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#text_format), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield), [weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weight)

**Description**: Indicates how the label should be rendered. The label includes the #title and the required marker, if #required. If the #title is empty but the field is #required, the label will contain only the required marker.

**Values**: String.

Possible values:

* **before**: The label is output before the element. This is the default for most elements.
* **after**: The label is output after the element. For example, this is used for radio and checkbox #type elements as set in system\_elements().
* **invisible**: #title is rendered as a label element before the form element in the page markup, and is made invisible with the Drupal 7 .element-invisible system CSS class ([system.base.css](https://api.drupal.org/api/drupal/7.x/search/system.base.css)). This makes #title remain available to screen-reader users, but hides it from being displayed visually in the browser.
* **attribute**: Set the title attribute on the element to create a tooltip but output no label element. This is supported only for checkboxes and radios in [form\_pre\_render\_conditional\_form\_element](https://api.drupal.org/api/drupal/includes%21form.inc/function/form_pre_render_conditional_form_element/7.x" \o "Adds form element theming to an element if its title or description is set.)(). It is used where a visual label is not needed, such as a table of checkboxes where the row and column provide the context. The tooltip will include the title and required marker.

**Usage example** ([block.module](https://api.drupal.org/api/drupal/7.x/search/block.module" \o "Multiple implementations exist.)):

<?php  
$form['blocks'][$key]['weight'] = [array](http://php.net/array)(  
 '#type' => 'weight',  
 '#default\_value' => $block['weight'],  
 '#delta' => $weight\_delta,  
 '#title\_display' => 'invisible',  
 '#title' => t('Weight for @block block', [array](http://php.net/array)('@block' => $block['info'])),  
);  
?>

### #tree

**Used by**: All

**Description**: Used to allow collections of form elements. Normally applied to the "parent" element, as the #tree property cascades to sub-elements. Use where you previously used ][ in form\_ calls. For more information, see [#tree and #parents](http://drupal.org/node/48643) in the handbook.

**Values**: TRUE or FALSE

**Usage example** ([system.module](https://api.drupal.org/api/drupal/7.x/search/system.module" \o "Multiple implementations exist.)):

<?php  
$form['status'] = [array](http://php.net/array)(  
  '#type' => 'checkboxes',   
  '#default\_value' => $status,   
  '#options' => $options,   
  '#tree' => TRUE,  
);  
$required = [array](http://php.net/array)('block', 'filter', 'system', 'user', 'watchdog');  
foreach ($required as $require) {  
  $form['status'][$require] = [array](http://php.net/array)(  
    '#type' => 'hidden',   
    '#value' => 1,   
    '#suffix' => t('required'),  
  );  
}  
?>

### #type

**Used by**: All

**Description**: Used to determine the type of form element.

**Values**: [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [checkbox](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkbox), [checkboxes](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#checkboxes), [date](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#date), [fieldset](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#fieldset), [file](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#file), [form](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#form), [hidden](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#hidden), [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#image_button), [item](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#item), [machine\_name](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#machine_name), [markup](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#markup), [password](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password), [password\_confirm](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#password_confirm), [radio](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radio), [radios](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#radios), [select](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#select), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit), [tableselect](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#tableselect), [textarea](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textarea), [textfield](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#textfield), [token](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#token), [value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#val), [vertical\_tabs](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#vertical_tabs), [weight](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#weight)

**Usage example** ([locale.module](https://api.drupal.org/api/drupal/7.x/search/locale.module" \o "Multiple implementations exist.)):

<?php  
$form['submit'] = [array](http://php.net/array)('#type' => 'submit', '#value' => t('Import'));  
?>

### #validate

**Used by**: [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "image_button), [form](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#form), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit)

**Description**: A list of custom validation functions that need to be passed.This is usually used to add additional validation functions to a form, or to use an alternate function rather than the default form validation function which is the form ID with \_validate appended to it.

**Values**: An array of function names. Each such function will take $form and $form\_state as parameters and should use [form\_set\_error](https://api.drupal.org/api/drupal/includes%21form.inc/function/form_set_error/7.x" \o "Files an error against a form element.)() if the form values do not pass validation. For example:

function test\_form\_validate($form, &$form\_state) {  
  if ($form\_state['values']['name'] == '') {  
    form\_set\_error('name', t('You must select a name for this group of settings.'));  
  }  
}

**Usage example** ([node.module](https://api.drupal.org/api/drupal/7.x/search/node.module" \o "Multiple implementations exist.)):

**Note:** If you are altering an existing form via [hook\_form\_alter](https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook_form_alter/7.x" \o "Perform alterations before a form is rendered.)() or a similar means, be careful with this property! You will probably want to add to the existing array rather than writing over it, so don't follow this usage example exactly.

<?php  
    // Node types:  
    $types = array\_map('check\_plain', node\_get\_types('names'));  
    $form['advanced']['type'] = [array](http://php.net/array)(  
      '#type' => 'checkboxes',  
      '#title' => t('Only of the type(s)'),  
      '#prefix' => '<div class="criterion">',  
      '#suffix' => '</div>',  
      '#options' => $types,  
    );  
    $form['advanced']['submit'] = [array](http://php.net/array)(  
      '#type' => 'submit',  
      '#value' => t('Advanced search'),  
      '#prefix' => '<div class="action">',  
      '#suffix' => '</div>',  
    );  
  
    $form['#validate'][] = 'node\_search\_validate';  
?>

### #validation\_arguments

INTERNAL

### #value

**Used by**: [button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#button), [hidden](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#hidden), [image\_button](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x" \l "image_button), [submit](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#submit), [token](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#token), [value](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#val)

**Description**: Used to set values that cannot be edited by the user. **Should NOT be confused with**[**#default\_value**](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7.x#default_value), which is for form inputs where users can override the default value.

**Values**: Mixed (text or numbers)

**Usage example** ([locale.module](https://api.drupal.org/api/drupal/7.x/search/locale.module" \o "Multiple implementations exist.)):

<?php  
$form['submit'] = [array](http://php.net/array)('#type' => 'submit', '#value' => t('Import'));  
?>

### #value\_callback

**Used by**: All elements.

**Description**: Specifies the name of a custom value function that implements how user input is mapped to an element's #value property.

**Values**: A single function name to be called to set the value of this element.

A value function for an element takes $element, $input and $form\_state as parameters, and has the form:

<?php  
function myelement\_value($element, $input = FALSE, $form\_state = [array](http://php.net/array)()) {  
  if ($input === FALSE) {  
    return [isset](http://php.net/isset)($element['#default\_value']) ? $element['#default\_value'] : 0;  
  }  
}  
?>

**Usage example** ([file.field.inc](https://api.drupal.org/api/drupal/7.x/search/file.field.inc)):

<?php  
  // Essentially we use the managed\_file type, extended with some enhancements.  
  $element\_info = element\_info('managed\_file');  
  $element += [array](http://php.net/array)(  
    '#type' => 'managed\_file',  
    '#default\_value' => [isset](http://php.net/isset)($items[$delta]) ? $items[$delta] : $defaults,  
    '#upload\_location' => file\_field\_widget\_uri($field, $instance),  
    '#upload\_validators' => file\_field\_widget\_upload\_validators($field, $instance),  
    '#value\_callback' => 'file\_field\_widget\_value',  
    '#process' => array\_merge($element\_info['#process'], [array](http://php.net/array)('file\_field\_widget\_process')),  
    // Allows this field to return an array instead of a single value.  
    '#extended' => TRUE,  
  );  
...  
function file\_field\_widget\_value($element, $input = FALSE, $form\_state) {  
  if ($input) {  
    // Checkboxes lose their value when empty.  
    // If the display field is present make sure its unchecked value is saved.  
    $field = field\_widget\_field($element, $form\_state);  
    if ([empty](http://php.net/empty)($input['display'])) {  
      $input['display'] = $field['settings']['display\_field'] ? 0 : 1;  
    }  
  }  
?>

### #weight

**Used by**: All elements.

**Description**: Used to sort the list of form elements before being output; lower numbers appear before higher numbers.

**Values**: A positive or negative number (integer or decimal).

**Usage example** ([book.module](https://api.drupal.org/api/drupal/7.x/search/book.module" \o "Multiple implementations exist.)):

<?php  
$form['parent'] = [array](http://php.net/array)(  
  '#type' => 'select',   
  '#title' => t('Parent'),   
  '#default\_value' => ($node->parent ? $node->parent : arg(4)),   
  '#options' => book\_toc($node->nid),   
  '#weight' => -15,  
  '#description' => t('The parent that this page belongs in. Note that pages whose parent is &lt;top-level&gt; are regarded as independent, top-level books.'),  
);  
?>

### File

developer/topics/forms\_api\_reference.html