Drupal features an application programming interface (API) for generating, validating, and processing HTML forms. The form API abstracts forms into a nested array of properties and values. The array is then rendered as part of the process when Drupal renders the page that contains the form. There are several implications of this approach:

- Rather than output HTML, we create an array and let the engine generate the HTML.
- Since we are dealing with a representation of the form as structured data, we can add, delete, reorder, and change forms. This is especially handy when you want to modify a form created by a different module in a clean and unobtrusive way.
- Any form element can be mapped to any theme function.
- Additional form validation or processing can be added to any form.
- Operations with forms are protected against form injection attacks, where a user modifies a form and then tries to submit it.

In this chapter, we'll face the learning curve head on. You'll learn how the form engine works; how to create forms, validate them, and process them; and how to pummel the rendering engine into submission when you want to make an exception to the rule. This chapter covers the form API as implemented in Drupal 7. We will start by examining how the form processing engine works. If you are just starting out with forms in Drupal and want to start with an example, you might want to jump ahead to the section titled “Creating Basic Forms.” If you are looking for details about individual form properties, you’ll find it in the last part of the chapter in the section titled “Form API Properties.”

**Understanding Form Processing**

Figure 11-1 shows an overview of the form building, validation, and submission process. In the following sections, we’ll be using this figure as a guide and describing what happens along the way.
Figure 11-1. How Drupal handles forms