jQuery UI IN ACTION

TJ VanToll

Foreword by Scott González

SAMPLE CHAPTER

MANNING

jQuery UI in Action

by TJ VanToll

Chapter 1

brief contents

Part 1

Meet jQuery UI

These first two chapters serve as an introduction to jQuery UI. As you'll learn in chapter 1, jQuery UI is a collection of plugins and utilities that build on jQuery, supported by the jQuery Foundation. You can count on them to be officially supported and maintained throughout the life of your application.

In chapter 1 you'll learn about the library itself—what's in it, who maintains it, what it does well, and even what it doesn't do well.

In chapter 2 you'll build on that knowledge to learn the ins and outs of widgets, the core building blocks of jQuery UI. The focus here is on three mechanisms the widget factory provides for customization: options, methods, and events. Options are configurable properties of widgets, methods let you perform actions on widgets, and events let you to respond to changes on the widgets.

What you learn about the library, and about the jQuery UI widgets, will give you the foundation you need to build more complex interfaces in part 2.

1 Introducing jQuery UI

This chapter covers

- What jQuery UI includes
- Whether jQuery UI is for you
- How to get started using the library

Let's take a trip back to early 2006. The term AJAX had been coined, the second beta of Internet Explorer 7 was released, and John Resig announced a small library he called jQuery. jQuery would soon become wildly popular, thanks in part to how easy it was to extend its core functionality through plugins.

Months passed, and thousands of plugins were created by the jQuery community. Although the abundance of plugins provided variety, they were scattered around the internet, had inconsistent APIs, and often had little or no documentation. Because of these problems, the jQuery team wanted to provide an official set of plugins in a centralized location. In September 2007 they created a new library with these plugins—jQuery UI.

From a high level, jQuery UI was, and still is, a collection of plugins and utilities that build on jQuery. But dig deeper and you find a set of consistent, well-documented, themeable building blocks to help you create everything from small websites to highly complex web applications.

Unlike jQuery plugins, the plugins and utilities in jQuery UI are supported by the jQuery Foundation. You can count on them to be officially supported and maintained throughout the life of your application.

The stability and ease of use of jQuery UI led to continuous growth in the library's popularity. The library is now used in 19% of the top 10,000 sites on the web, and has been incorporated into WordPress core and Drupal.

In this book you'll learn how to use the pieces of jQuery UI to create powerful and interactive websites and applications. In this chapter you'll start by taking a thorough look at what the jQuery UI library is, why you'd want to use it, and how to download the library and get it up and running. Let's get started!

Who is this book for?

This book assumes that you have basic knowledge of CSS, JavaScript, and jQuery. If you're not an expert don't despair—when intermediate- and advanced-level concepts are brought up, they're explained. If you're finding yourself a bit overwhelmed, appendix A discusses resources for getting up to speed. On the flip side, if you're an expert don't despair either. We'll build a number of real-world examples and discuss advanced aspects of the library throughout the book.

1.1 What is in jQuery UI?

The plugins and utilities in jQuery UI are divided into four categories—widgets, interactions, effects, and utilities (the structure of the library is presented in figure 1.1):

- *Widgets* are jQuery plugins used to create UI elements such as datepickers and menus. As of version 1.11, the library has 12 widgets, shown in figure 1.2. The widgets in jQuery UI adhere to the library's CSS framework, and therefore have a consistent look and feel. We'll cover the jQuery UI widgets in chapters 2, 3, and 4 and the CSS framework in chapter 7.
- *Interactions* are jQuery plugins that give the user the ability to interact with DOM elements. The draggable interaction allows users to drag elements around the screen, and the sortable interaction allows users to sort items in a list. We'll cover interactions in chapter 5.
- *Effects* are a full suite of custom animations and transitions for DOM elements. They're built on the animations provided in jQuery Core, and enhance a number of Core's methods such as `show()` and `hide()`. We'll cover effects in chapter 6.
- *Utilities* are a set of modular tools the library uses internally. The widget factory is the mechanism all jQuery UI widgets are built with; we'll cover it in chapters 8 and 9. The position utility provides an easy and precise means of positioning elements on the screen. We'll cover position and the rest of the utilities in jQuery UI in chapter 12.

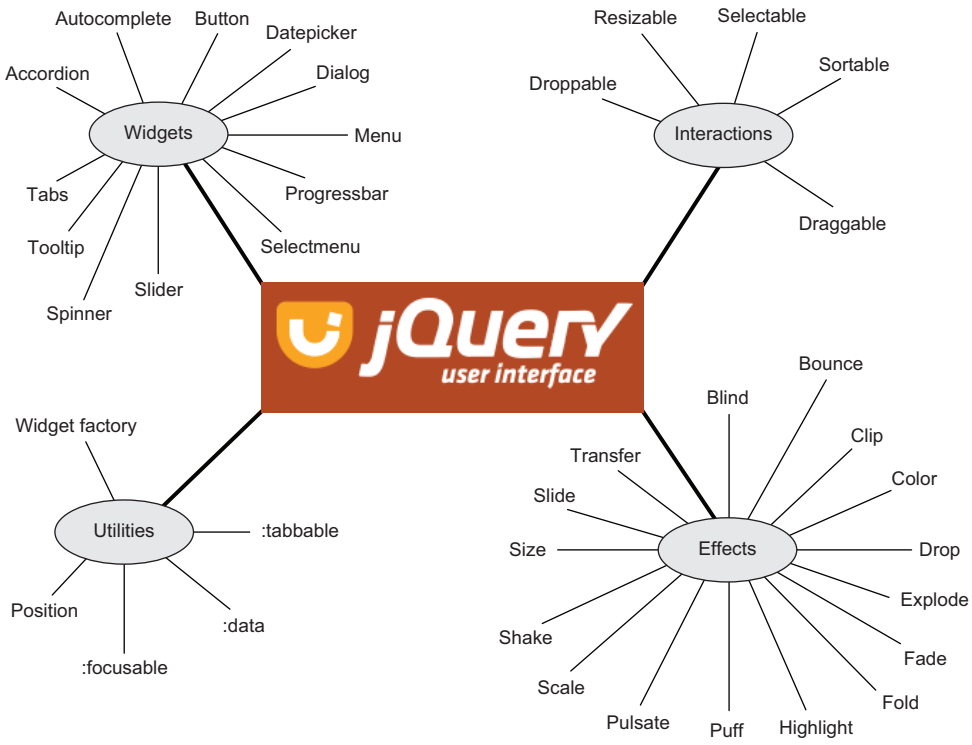

Figure 1.1 **The pieces of the jQuery UI library, categorized into widgets, interactions, utilities, and effects**

The pieces of jQuery UI work well together, but they were also designed with modularity in mind. Although the widget factory and position utility are heavily used in the library, they're also standalone plugins that can be used outside of jQuery UI; their only dependency is jQuery Core.

Now that we've seen what jQuery UI includes, let's see what jQuery UI can be used for, and how it might be a good fit for your next project.

Who is jQuery UI?

Development on jQuery UI (as well as all jQuery projects) is coordinated by the jQuery Foundation—a nonprofit association funded by community contributions of time and money.

The jQuery UI team is a group of eight individuals (I am one of them) scattered throughout the world. I became enthralled with jQuery UI after I discovered the amazing number of things the library could do with a small amount of code. I started submitting bug fixes and documentation and haven't looked back.

I hope you become as excited about the library as I am. The jQuery UI project is primarily community and volunteer driven, and there's always plenty to do!

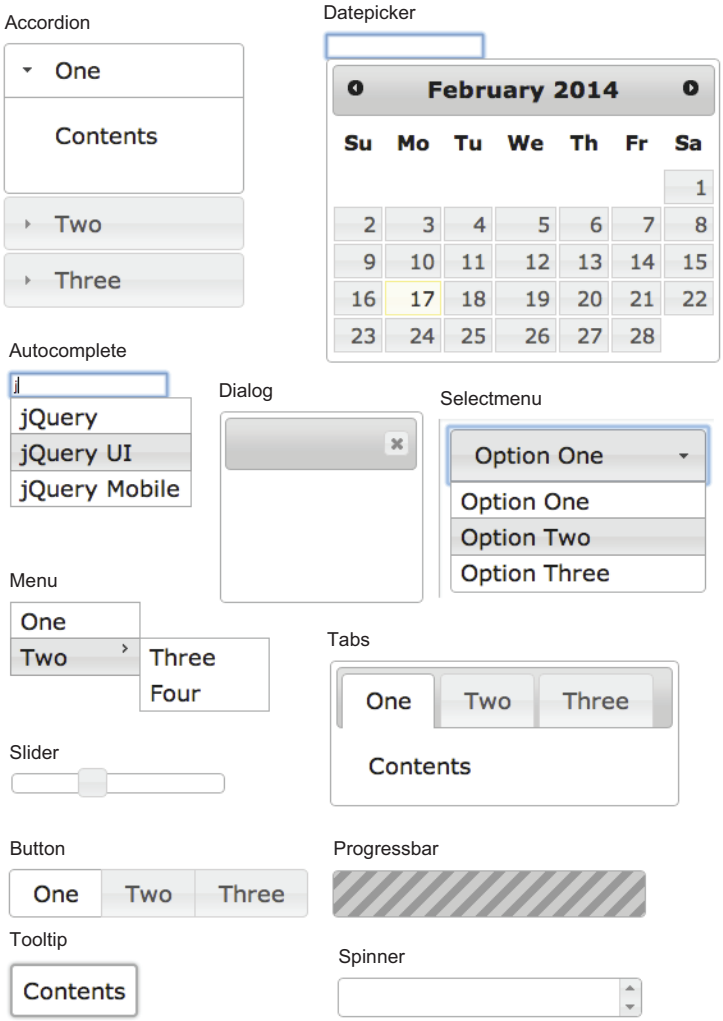

Figure 1.2 An example of all 12 jQuery UI widgets. Because of the jQuery UI CSS framework, each widget has a consistent look.

1.2 The benefits of using jQuery UI

Any website or application that uses jQuery almost certainly has a use for jQuery UI. jQuery Core is powerful, but it's a small library that doesn't do everything you need to build modern web applications. If you've been frustrated by searching the internet and piecing together jQuery plugins, then jQuery UI provides an appealing alternative. Let's look at the advantages of using the library.

1.2.1 Cohesive and consistent APIs

Because jQuery plugins have different authors, they often have wildly inconsistent APIs. jQuery UI has also faced this problem. The jQuery UI library started as a collection of popular plugins by numerous authors with a variety of programming styles. This resulted in years of refactoring to present a consistent API to end users.

Throughout the process, common patterns emerged and were abstracted into utilities like the widget factory.

Because jQuery UI provides consistent APIs, users can move from one part of the library to another without constantly needing to refer to online documentation.

1.2.2 Comprehensive browser support

When using jQuery UI, you can feel confident that your code works in all major browsers. As of version 1.11, jQuery UI supports Internet Explorer versions 7 and up, as well as the latest two versions of Chrome, Firefox, Safari, and Opera. With jQuery UI, you write your code once and it runs everywhere.

> **NOTE** Internet Explorer 6 support was dropped in version 1.10 of jQuery UI due to low global usage. If you still need Internet Explorer 6 support, you can use version 1.9 of jQuery UI.

1.2.3 Open source and free to use

Everything in jQuery UI is open source. The library's source files are publicly available at https://github.com/jquery/jquery-ui. Not only are the source files open source but the project's home page and API documentation are as well (see https://github.com/jquery/jqueryui.com and https://github.com/jquery/api.jqueryui.com, respectively).

All development is done in the open, and the community is encouraged to participate. If you find a bug in the library, you can submit a patch for it. If you're confused by the documentation, you can ask for clarification. If you find a typo, you can submit a patch that fixes it. The development of all jQuery projects is community driven, and contributions are always welcome. For more information on contributing to jQuery, see appendix E.

jQuery UI is also free. The use of jQuery UI (and all jQuery projects) is under the terms of the MIT license. All jQuery projects are free to use in any project (including commercial ones), as long as the copyright headers are preserved.

1.2.4 Thorough documentation

One of the major pain points with jQuery plugins is the difficulty of finding up-to-date and accurate documentation. All pieces of jQuery UI are thoroughly and consistently documented at http://api.jqueryui.com/. By default, the APIs for the latest version are shown, but previous versions are available as well. For example, http://api.jqueryui.com/1.10/ shows the APIs for 1.10 and http://api.jqueryui.com/1.9/ shows the APIs for 1.9.

1.2.5 Powerful theming mechanism

Another challenge of working with plugins is creating a consistent look. Although some plugins provide a way to theme the elements they create, the conventions used are often wildly different. jQuery UI solves this with a CSS framework that all its widgets use; therefore, all widgets look the same out of the box, but you still have the flexibility to create your own custom look and feel.

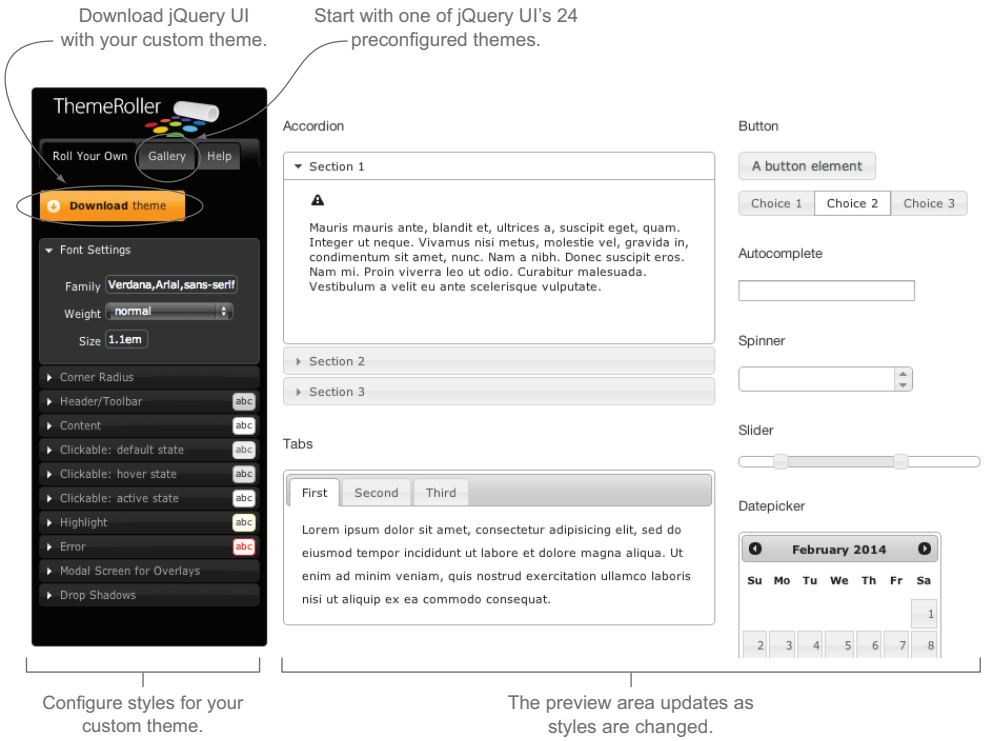

Figure 1.3 Using jQuery UI ThemeRoller, you can configure a custom theme by playing with CSS properties and seeing their effect on the jQuery UI widgets live.

To make this process easier, the jQuery UI ThemeRoller allows you to visually play with the widgets' displays and generate a CSS file with your theme. Not a designer? No worries. jQuery UI also provides 24 themes you can use or build on top of. ThemeRoller is available at http://jqueryui.com/themeroller/ and is shown in figure 1.3.

1.2.6 Emphasis on accessibility

Accessibility is an important consideration when building anything for the web, but making even simple applications accessible to all audiences can be a difficult task. Documentation is scarce, screen readers can be tricky to test on, and specifications such as Accessible Rich Internet Applications (ARIA) can be complex and difficult to understand.

All jQuery UI widgets are designed with accessibility in mind. You can add widgets to your site and feel confident that everyone can use them. The jQuery UI widgets are keyboard accessible, use ARIA roles appropriately, and use proper markup to optimize user experiences on screen readers.

NOTE ARIA is a technical specification published by the World Wide Web Consortium (W3C). It aims to improve the accessibility of web pages—specifically pages with dynamic content and UI components. It specifies a number of HTML attributes that can be applied to elements to help assistive technologies such as screen readers interpret web pages.

1.2.7 Stable and maintenance friendly

Because jQuery UI is maintained by the jQuery Foundation, the library is updated as new versions of jQuery Core and browsers are released. Although using the latest version of the library is encouraged, the jQuery UI team realizes the difficulty of upgrading large and complex applications.

Therefore, two versions of the library are maintained simultaneously. Fixes made to the latest stable release can be incorporated in the previous legacy release. APIs are never removed from the library without being deprecated for a full major release.

To help with upgrading, a detailed guide is published with each major release of the library. The upgrade guide for 1.11 is at http://jqueryui.com/upgrade-guide/1.11/, and the upgrade guide for 1.10 is at http://jqueryui.com/upgrade-guide/1.10/.

A changelog, listing every change—including bug fixes—made to the library in that release, is also produced. The changelog for 1.11.0 is at http://jqueryui.com/changelog/1.11.0/, and the changelog for 1.10.4 is at http://jqueryui.com/changelog/1.10.4.

Now that you know why you'd want to use jQuery UI, let's discuss why you might not want to use the library.

1.3 The limitations of jQuery UI

Although jQuery UI solves a lot of problems, it doesn't solve everyone's. The library receives two main complaints: it doesn't have enough widgets, and it's not optimized for mobile. Let's deal with each of these.

1.3.1 Lack of widgets

As of version 1.11, jQuery UI has 12 widgets. Although these widgets are in the library because they solve common UI problems, 12 widgets certainly don't solve every UI problem that even a small company encounters.

Fortunately, you can use jQuery UI alongside community and commercially written jQuery plugins. Many third-party plugins use portions of jQuery UI, such as the widget factory and the CSS framework, to provide a consistent API and a consistent theme.

If you can't find a widget to meet your needs, it's easy to build your own with jQuery UI. We'll discuss how to build custom widgets using the widget factory in chapter 8.

Finally, all jQuery UI widgets are built with extensibility in mind. You can make subtle alterations to the library's widgets or build completely new widgets on top of them easily. We'll discuss extending jQuery UI widgets in chapter 9.

1.3.2 jQuery UI and mobile devices

The other major complaint about jQuery UI is that the library isn't optimized for mobile devices. The primary issues cited are the lack of touch-event support, the display of the widgets, and the size of the library. Let's tackle each of these individually:

- *Touch-event support*—As of version 1.11, jQuery UI doesn't natively support touch events. By default, some widgets and interactions don't work on mobile browsers such as iOS Safari or Chrome for Android. But a workaround is available until true support for touch events comes in a future release. We'll discuss the issues with touch events, how to get jQuery UI to work with them, and future plans for true support when we discuss interactions in chapter 5.
- *Display of widgets*—The look and feel of jQuery UI widgets are more suited for desktop browsers than mobile ones. To address this, the jQuery UI team is working with the jQuery Mobile team to build widgets that look good on all screen sizes. In the meantime, because all jQuery UI widgets conform to the jQuery UI CSS framework, it's easy to adjust the display of all widgets to meet your needs. We'll discuss the jQuery UI CSS framework, along with specific mobile considerations, in chapter 7.
- *Size of the library*—File size is important for any client-side library, especially on mobile devices where connection speed can be limited and latency is frequently high. jQuery UI is a large library with many components, and the full library is a lot to download. But jQuery UI is modularly written, so it's easy to create a build with only the pieces of the library that you need. Although creating a custom build is important for any site or application, it's vital if you're targeting mobile devices. We'll discuss custom builds in chapter 10.

If you're building a site or application that *solely* targets mobile devices, you should consider a mobile-centric framework like jQuery Mobile. But if you're building for desktop and mobile, you can still get all the benefits of jQuery UI with a few tweaks to optimize the mobile experience, which we'll discuss throughout the book.

Now that we've looked at the advantages and limitations of jQuery UI, let's look at how to use it.

jQuery UI vs. jQuery Mobile

jQuery Mobile is a UI framework that creates experiences that work on all devices. Like jQuery UI, jQuery Mobile is a series of widgets and utilities built on jQuery Core. In fact, jQuery Mobile includes the jQuery UI widget factory and uses it to create all its widgets.

Because of the similarity in the two frameworks, the teams are working to merge the common pieces of the projects. The end goal is a single set of widgets that work on any device. As a first step, jQuery Mobile's 1.4 release included the jQuery UI tabs widget. This collaboration continuously improves the mobile device support in jQuery UI.

1.4 Getting started with the library

You can get a copy of jQuery UI two ways: download the library from http://jqueryui.com/ or retrieve the files from a content delivery network (CDN). You'll learn about each of these options, but first you need to decide what version of the library to use.

1.4.1 Versions of the library

In this book we'll cover *version 1.11* of jQuery UI. The final position in the version number (1.11.*1*, 1.11.*2*, and so on) is reserved for bug fix releases. Because breaking changes are never introduced in bug fix releases, you can use any release in the 1.11 series with the examples in this book. The code examples explicitly use 1.11.0, but the latest bug fix release in the 1.11 series is recommended.

What's new in jQuery UI 1.11?

The two main features of jQuery UI 1.11 are a new widget, selectmenu, and complete Asynchronous Module Definition (AMD) support to use for dependency management.

Selectmenu is an accessible, customizable, and themeable replacement for the native `<select>` element. You'll learn how to use selectmenu, as well as the other widgets jQuery UI provides for building forms, in chapter 3.

AMD allows you to create highly customized builds of jQuery UI so that users download only the portion of the library that they need. We'll look at AMD when we discuss custom builds and preparing the library for production in chapter 10.

1.4.2 Downloading from the jQuery UI website

The first of the two options is downloading the library from http://jqueryui.com. There you'll find the download section shown in figure 1.4.

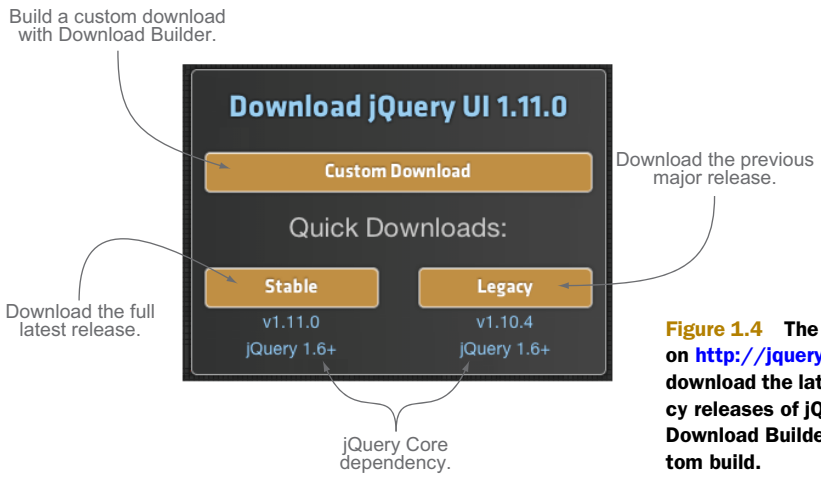

Figure 1.4 The download prompt on http://jqueryui.com. You can download the latest stable or legacy releases of jQuery UI, or visit Download Builder to create a custom build.

Let's look at each element of the download prompt:

- *Build a custom download with Download Builder*—The Custom Download button links to the jQuery UI Download Builder. Download Builder allows you to create a custom build that includes only the portions of the library that you need. This is ideal for production, as you want users to download only the portions of the library they need. For development, it's convenient to have the entire library available, and therefore you won't build a custom download for now. You'll build a production version of the library in chapter 10.
- *Download the latest release*—The quick downloads are links to zip files containing all the files in the library. The Stable button links to a zip file with the files for the latest released version.
- *Download the previous major release*—The Legacy button links to a zip file with all the library's files, but for the previous major version of the library (recall that two versions are maintained simultaneously).
- *jQuery Core dependency*—To aid users in upgrading, jQuery UI maintains compatibility with multiple versions of jQuery Core. Both versions 1.10.x and 1.11.x can be used with any version of jQuery Core 1.6 or higher.

The zip files downloaded using the Stable or Legacy buttons contain every file you need, including all dependencies. Although it's helpful to have all these files when preparing an application for production, it can be overwhelming when getting started. There's an easier way to get the library up and running.

1.4.3 Downloading from CDNs

A content delivery network (CDN) is a network of servers designed to serve files to users. Using a CDN moves the responsibility of hosting files from your own servers to a series of external ones. The jQuery Foundation, Google, and Microsoft all provide CDNs that host jQuery Core as well as jQuery UI. You can find documentation and a full listing of the libraries each host provides at the following URLs:

- *jQuery*—http://code.jquery.com/
- *Google*—https://developers.google.com/speed/libraries/devguide
- *Microsoft*—http://www.asp.net/ajaxlibrary/cdn.ashx

Because a CDN doesn't require you to host your own version of jQuery and jQuery UI, it's perfect for demos and experimentation. You'll use CDN versions of the library throughout this book. Next, you'll learn how to take these files from a CDN and get them on a web page.

1.5 The first example

You've seen how to download jQuery UI. Now let's see how you can use it. You need to build an HTML page that includes jQuery Core, jQuery UI's CSS, and jQuery UI's JavaScript.

All examples in this book use the same boilerplate HTML using jQuery's CDN (http://code.jquery.com) to download all jQuery files. The boilerplate is shown in the following listing.

Listing 1.1 Boilerplate for examples

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>First Demo</title>
    <link rel="stylesheet"
href="http://code.jquery.com/ui/1.11.0/themes/smoothness/jquery-ui.css">
</head>
<body>
    <!-- Your HTML here -->

   <script src="http://code.jquery.com/jquery-1.11.1.js"></script>
    <script src="http://code.jquery.com/ui/1.11.0/jquery-ui.js"></script>

    <!-- Your JavaScript here -->
</body>
</html>
```

An HTML5 doctype. jQuery Core and UI only support standards mode. This doctype puts all browsers in standards mode.

Import version 1.11.0 of jQuery UI's style sheet from jQuery's CDN.

Import version 1.11.1 of jQuery Core from jQuery's CDN.

Import version 1.11.0 of jQuery UI's JavaScript from jQuery's CDN.

The placement of the style sheet and scripts is important. Style sheets are placed in the `<head>` of the document so that HTML elements in the `<body>` are styled as they're rendered. When style sheets are placed after elements in the `<body>`, the user may experience a flash of unstyled content (FOUC). In this case, elements are rendered without styling, and subsequently enhanced after the style sheet is downloaded and parsed by the browser.

Conversely, scripts are placed last in the `<body>`, after any HTML the page needs. This is done for two reasons. First, if something were to go wrong with the download, parsing, or execution of the script, or if the user had JavaScript disabled, the content of the web page would still be available to the user. Second, because the scripts are at the end of the page, any JavaScript you write doesn't depend on whether the DOM is ready.

The examples in this book assume that the boilerplate shown in listing 1.1 is in place, and the `<!—Your HTML here -->` and `<!—Your JavaScript here -->` comments indicate where you insert content. Here's an example of a jQuery UI datepicker:

```
<input id="datepicker">
<script>
    $( "#datepicker" ).datepicker();
</script>
```

Waiting for the DOM to be ready

Historically, `<script>` tags have been placed in the `<head>` of HTML documents. When the browser executes these scripts, the `<body>` isn't rendered. Therefore, scripts need to wait for the browser's `DOMContentLoaded` event before they can access DOM elements. jQuery Core provides a shorthand for doing this:

```
$(function() {
    // The DOM is now ready.
});
```

When scripts are placed at the end of the document (before `</body>`), the wrapping `$(function() {} )` is no longer necessary.

The following listing shows the example after the datepicker code has been inserted into the boilerplate.

Listing 1.2 First example: building a datepicker

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>First Demo</title>
    <link rel="stylesheet" href="http://code.jquery.com/ui/1.11.0/themes/
     smoothness/jquery-ui.css">
</head>
<body>
    <input id="datepicker">

    <script src="http://code.jquery.com/jquery-1.11.1.js"></script>
    <script src="http://code.jquery.com/ui/1.11.0/jquery-ui.js"></script>

    <script>
        $( "#datepicker" ).datepicker();
    </script>
</body>
</html>
```

Save this text as a .html file, and open it in a browser. Give the input focus, and you see the datepicker shown in figure 1.5.

That's it. With one line of HTML and one line of JavaScript, you have a fully functional datepicker!

The full source code for the examples presented throughout this book is available for download at https://github.com/tjvantoll/jquery-ui-in-action-demos. You don't have to keep

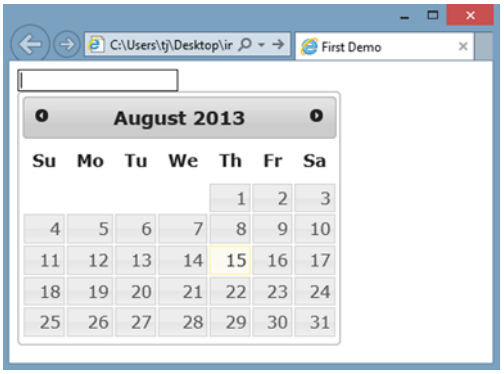

Figure 1.5 The first example. A jQuery UI datepicker opens when the `<input>` receives focus.

track of the boilerplate in your head. The datepicker code can be found at chapter01/01-building-a-datepicker.html.

But there's an even easier way to play with jQuery UI—without having to leave your browser.

jQuery coding standards

You can write an expression such as `$( "#datepicker" )` in JavaScript in several ways: `$( '#datepicker' )`, `$("#datepicker")`, or `$('#datepicker')`. jQuery UI as well all jQuery projects consistently follow jQuery's JavaScript style guide (http://contribute.jquery.org/style-guide/js/).

For consistency, this book adheres to the conventions in this guide. Notable conventions include using double quotes for strings (`"jQuery"` and not `'jQuery'`) and the liberal use of spacing—`$( "#datepicker" )` and not `$("#datepicker")`. These are jQuery's internal conventions and not requirements of projects using jQuery. If you prefer single quotes then use them. The most important thing is to be consistent in your own usage; don't use single quotes in one function and double quotes in the next.

1.6 Using an online testing tool

Online testing tools allow you to write HTML, CSS, and JavaScript in the browser and preview the results live. You can also save examples and get a unique URL you can save or share with others. You'll use these tools to set up your boilerplate and save it in a bookmark.

JS Bin (http://jsbin.com/), jsFiddle (http://jsfiddle.net), and CodePen (http://codepen.io/) are examples of these services. Although the core functionality of each service is roughly the same, each has unique features, and you can play with them to see which you like best. Let's look at how to run your datepicker example in jsFiddle.

Visit http://jsfiddle.net. The pertinent portions of the UI are shown in figure 1.6.

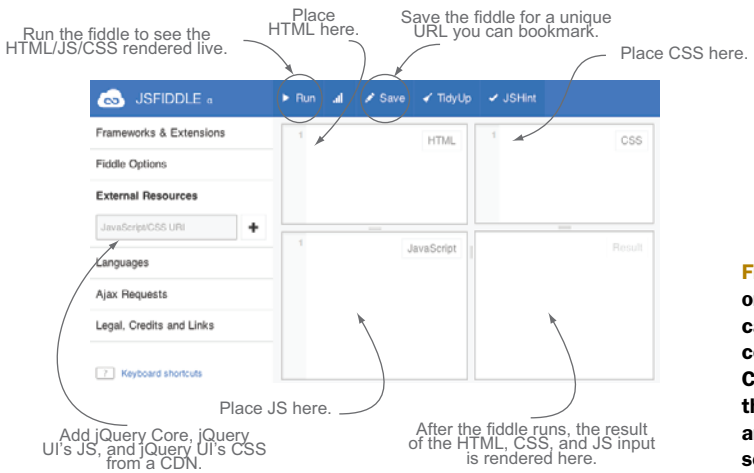

Figure 1.6 jsFiddle is an online testing tool that you can use to run jQuery UI code. You place HTML, CSS, and JavaScript in their appropriate panes, and click the Run button to see the results.

First, you need to make jQuery and jQuery UI available as external resources. The URLs you want to use are

- http://code.jquery.com/ui/1.11.0/themes/smoothness/jquery-ui.css
- http://code.jquery.com/jquery-1.11.1.js
- http://code.jquery.com/ui/1.11.0/jquery-ui.js

You can copy and paste these URLs from http://code.jquery.com if you want to avoid typos or to play with other versions. After you add the resources, save the fiddle. This saves the current state and gives you a unique URL you can bookmark so you don't have to enter the external resources again. After this setup, you can enter HTML, JavaScript, and CSS. Then, run the example, and the result displays in the Result pane.

Because the datepicker is one line of HTML and one line of JavaScript, to run the example in jsFiddle you place those lines in the appropriate panes and run the fiddle. The result is shown in figure 1.7.

> **NOTE** You can view this example live at http://jsfiddle.net/tj_vantoll/Eda2W/. If you append /show to the end of a jsFiddle URL (for instance, http://jsfiddle.net/tj_vantoll/Eda2W/show/), you can view the example outside of the jsFiddle UI—it's the equivalent of looking at just the Result pane. Finally, if you create a jsFiddle account, you can use http://jsfiddle.net/draft/ to view the result of last example you ran. Because the draft URL is short (and bookmarkable), it's handy for testing on mobile devices.

jsFiddle handles the boilerplate for you so you can concentrate on jQuery UI, making it a convenient option for playing with the examples provided throughout this book.

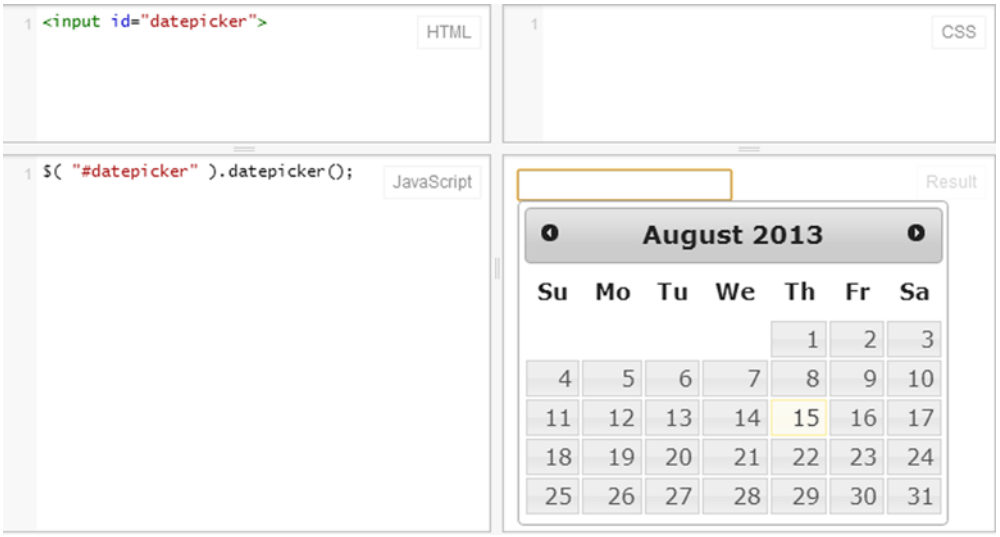

Figure 1.7 The datepicker example running in jsFiddle. The jsFiddle interface takes the HTML in the HTML pane and the JavaScript in the JavaScript pane, runs them, and displays the results in the Result pane.

1.7 Summary

jQuery UI is a collection of widgets, effects, interactions, and utilities to help you build powerful websites and applications. jQuery UI is known for its stable, cohesive APIs, excellent browser support, and comprehensive documentation.

You can download jQuery UI from http://jqueryui.com or from a CDN. You can test jQuery UI locally or use an online testing tool such as JS Bin, jsFiddle, or CodePen. You saw how easy it is to build powerful UI elements by creating a datepicker with one line of HTML and one line of JavaScript.

In the next chapters, you'll explore the functionality that the jQuery UI library provides. You'll start in chapter 2 with a deeper look at the core components of jQuery UI: widgets.

WEB DEVELOPMENT

jQuery UI IN ACTION

TJ VanToll

You're only one tag away from richer user interfaces—<script src="jquery-ui.js">. The jQuery UI library simplifies web UI development by providing robust widgets, interactions, and effects you can use immediately. It includes datepickers, autocompletes, tooltips, and a whole lot more. And, jQuery UI's powerful widget factory makes it a snap to customize existing components to meet your needs.

jQuery UI in Action is a practical guide to using and customizing jQuery UI library components. By working through numerous examples, you'll quickly master jQuery UI's twelve widgets and five interactions—draggable, droppable, resizable, selectable, and sortable. The engaging examples illustrate techniques that work across all devices. You'll use the widget factory to create reusable plugins and discover jQuery UI's CSS theming system that allows you to create a custom, cohesive look for your sites and your applications.

What's Inside

- Create interactions that work on any device
- Customizable widgets for web and mobile apps
- Written by a member of the core jQuery UI team
- Covers jQuery UI 1.11

Written for front-end developers and web designers with a basic understanding of jQuery.

A professional web developer, **TJ VanToll** is a member of the jQuery UI core team.

To download their free eBook in PDF, ePub, and Kindle formats, owners of this book should visit manning.com/jQueryUIinAction

"A fantastic resource."
—From the Foreword by Scott González, Project Lead, jQuery UI

"A complete and detailed guide to writing web user interfaces."
—Gregor Zurowski, Sotheby's

"Excellent, in-depth explanations of jQuery UI's inner workings."
—Linda Carver
Wicked Coursing LLC

"Articulate, well-organized, easy to read, and thorough."
—Philip Taffet, SOHOsoft LLC

ISBN 13: 978-1-617291-93-7
ISBN 10: 1-617291-93-5
54499
9 781617 291937

MANNING $44.99 / Can $47.99 [INCLUDING eBOOK]