



Learn by doing: less theory, more results

jQuery UI Themes

Create new themes for your jQuery site with this step-by-step guide

Beginner's Guide

Adam Boduch

[PACKT] open source 
PUBLISHING community experience distilled

jQuery UI Themes

Beginner's Guide

Create new themes for your jQuery site with this step-by-step guide

Adam Boduch

[PACKT] open source 
PUBLISHING community experience distilled

BIRMINGHAM - MUMBAI

jQuery UI Themes

Beginner's Guide

Copyright © 2011 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: July 2011

Production Reference: 1160711

Published by Packt Publishing Ltd.
32 Lincoln Road
Olton
Birmingham, B27 6PA, UK.

ISBN 978-1-849510-44-8

www.packtpub.com

Cover Image by Asher Wishkerman (a.wishkerman@mpic.de)

Credits

Author

Adam Boduch

Project Coordinator

Joel Goveya

Reviewers

Md. Mahmud Ahsan

Nick Gajewski

Shameemah Kurzawa

Joe Wu

Proofreader

Aaron Nash

Indexer

Hemangini Bari

Acquisition Editor

Sarah Cullington

Graphics

Nilesh R. Mohite

Valentina J. D'silva

Development Editor

Gaurav Mehta

Production Coordinator

Aparna Bhagat

Technical Editors

Gauri Iyer

Pooja Pande Malik

Cover Work

Kruthika Bangera

Aparna Bhagat

About the Author

Adam Boduch has been programming in Python for nearly a decade. He is experienced in working with several web frameworks such as Django and Twisted. He likes to experiment with integrating JavaScript tools such as jQuery UI into these frameworks.

Adam has been working for Enomaly Inc. since 2006. He started working with content management systems before making the transition to ECP, where he designed several user interface components using jQuery UI. He now leads the SpotCloud cloud computing market-place project.

I'd like to thank Melissa and Jason for their endless love and support, without which, this book would not have been possible.

About the Reviewers

Md. Mahmud Ahsan graduated in Computer Science & Engineering from the International Islamic University Chittagong (IIUC) in Bangladesh. He is a Zend Certified engineer and has experience of more than six years in LAMP-based web applications development. He is an expert in developing APIs and mashup applications in Facebook, LinkedIn, and Twitter. Besides his full-time freelance work, he blogs at <http://thinkdiff.net> and also writes articles in different technologies such as Facebook application development. He lives in Bangladesh with his wife, Jinat.

Currently, he is working as a freelancer, managing and developing social web applications, using technologies such as PHP, MySQL, JavaScript, Zend Framework, CodeIgniter, jQuery, and Mashup APIs. He also leads various small- to medium-level projects.

He is also an Indie iPhone application developer and publishes the applications he develops at <http://ithinkdiff.net>.

He has worked as a technical reviewer on *Zend Framework 1.8 Web Application Development* and *PHP jQuery Cookbook* by Packt Publishing.

I am very grateful to my father, who bought me a computer in 2001. Since then, I was able to explore my love for programming and work in various technologies.

Nick Gajewski is a user interface and Drupal developer. He lives in Toronto, Ontario, Canada.

Originally tinkering with websites during the years of the Internet Explorer and Netscape browser wars, he took a partial hiatus from the world of the Internet to educate the youth of today.

After five years in the education trenches, he was lured back into web development and design through its innovation, creativity, and limitless possibilities. Now, with four years of experience under his belt, he builds front-end and offers Drupal solutions for Enomaly Inc. and freelances under `nickgajewski.com`.

He enjoys creating websites that are exciting, innovative, and are a pleasure for the users to experience.

I would like to thank my family for their support, encouragement, humor, and delicious Polish food.

Shameemah Kurzawa started programming when she was in high school. Being motivated to be a System Analyst, she pursued both undergraduate and postgraduate studies in Business Information System and Software Engineering, respectively.

She has been working as a Web Developer/Analyst for the past five years at a renowned broadcasting company in Australia, SBS (Special Broadcasting Service). Besides work, she enjoys spending her time with family. She is a mother of a little baby boy, aged two. She also enjoys travelling, cooking, as well as reading about new technologies.

I would like to thank my husband, my son, and the Packt Publishing team for the support and understanding in reviewing this book.

Joe Wu is a full-time Senior PHP Web Developer, and has been in the industry since 2005. He has worked on various projects of all sizes, and is familiar with most of the Open-Source technologies surrounding PHP Web Development.

Joe is always enthusiastic about new and upcoming technologies, and is keen to learn and pick up a new skill-set wherever possible and utilize them in his current or future projects. He is also keen on learning about new opportunities and innovative ideas, and believes that the market is always wide open for new and upcoming innovations to improve our ways of living.

Apart from all the technological computer work, Joe is a professional Badminton Player and manages to fit a near full-time training schedule along with his full-time job. Joe's best ranking of 59th in the world in singles and the attendance of Commonwealth Games Delhi 2010 means that he has equally as much experience in Badminton and Web Developing.

Apart from all the endeavors, Joe also works for his own company (with his business partner) to put his skills and experience to good use and help anyone who needs assistance with web development.

Wackyinnovation (www.wackyinnovation.com) promotes the concept of always moving forward and coming up with and utilizing new technologies and ideas. Their always enthusiastic and can-do attitude ensures jobs are done to perfection with an innovative edge to their competitors.

I would like to thank everyone around me for their continued support both towards badminton and work, especially my fiancé who has to put up with my endless endeavors and dreams that I am pursuing.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- ◆ Fully searchable across every book published by Packt
- ◆ Copy and paste, print and bookmark content
- ◆ On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	1
Chapter 1: Themeable User Interfaces	7
What is a theme?	7
Themes in the real world	8
Desktop themes	8
Themes on the Web	8
What are widgets?	9
Widget structure	9
Widgets and themes	10
Styling user interfaces	10
Time for action - creating a style	10
Style colors	13
Style fonts	13
Theme basics	14
Time for action - grouping styles	14
Repetitive style properties	16
Group selectors	16
Time for action - nesting styles	17
Nested selectors	19
Nesting depth	19
Theme layout	20
Time for action - placing elements on the page	20
Layout design	23
Consistent layout	23
jQuery UI themes	24
The widget framework	24
The CSS framework	25
Theme-ready widgets	25
Summary	25

Chapter 2: Using Themes	27
Downloading jQuery UI	27
Time for action - building a download	27
Minimal downloads	29
Time for action - removing downloaded components	29
Theme structure	30
Development bundle	30
File structure	31
Class structure	32
Using jQuery UI	33
Time for action - using widgets	33
Widget options	36
Widget methods	37
Widgets and the DOM	37
Widgets and CSS	38
Switching themes	38
Time for action – exploring the theme switcher widget	39
Time for action - themeroller dev tool	41
Other frameworks	43
Dojo	43
Ext JS	44
Summary	44
Chapter 3: Using the ThemeRoller	47
ThemeRoller basics	47
Instant feedback	48
The dev tool	48
Portability	48
ThemeRoller gallery	49
Viewing themes	49
Time for action - previewing a theme	49
Downloading themes	50
Time for action - downloading a theme	50
Themes from scratch	52
Fonts	53
Time for action - setting theme fonts	53
Corners	54
Time for action - changing the corner radius	55
Headers	56
Time for action - setting theme headers	56
Content	58

Time for action - setting widget content styles	58
States	60
Time for action - setting default state styles	60
Time for action - setting hover state styles	61
Time for action - setting active state styles	62
Cues	64
Time for action - changing the highlight cue	65
Time for action - changing the error cue	66
Overlays and shadows	67
Time for action - dialog overlays	67
Time for action - defining shadows	68
Changing existing themes	71
Selecting a theme	71
Time for action - adjusting a theme	71
Summary	73
Chapter 4: Working with Widget Containers	75
What are widget containers?	75
Containers and widgets	76
The ui-widget class	76
Time for action - preparing the example	77
Widget fonts	78
Time for action - changing widget fonts	78
Scaling widget fonts	79
Time for action - scaling down font size	80
Widget form fields	81
Time for action - changing widget form fields	82
The ui-widget-content class	85
Content borders	85
Time for action - styling content borders	85
Time for action - border sides	87
Content links	88
Time for action - changing link colors	89
The ui-widget-header class	91
Header borders	92
Time for action - Styling header borders	92
Time for action - border sides	94
Header links	95
Time for action - header links	96
Summary	98

Chapter 5: Transforming Interaction States	99
What are interaction states?	99
Time for action - preparing the example	100
The default state	101
Default state selectors	102
Time for action - default container selectors	102
Time for action - default link selectors	105
Default state borders	108
Time for action - default border styles	108
Default state font	111
Time for action - default font styles	111
Default state background	114
Time for action - default background styles	115
The hover state	118
Hover state selectors	118
Time for action - hover container selectors	118
Hovering and focusing	121
Time for action - separating the hover and focus states	121
Hover state font	122
Time for action - hover font styles	123
The active state	126
Active state selectors	126
Time for action - active container selectors	126
Active state background	128
Time for action - active background styles	129
Summary	131
Chapter 6: Customizing Interaction Cues	133
What are interaction cues?	133
Time for action - preparing the example	134
The highlight state	135
Highlight borders	136
Time for action - highlight message borders	136
Time for action - highlight button borders	138
Highlight background	139
Time for action - highlight message background	139
Time for action - highlight date-picker background	141
Highlight font	142
Time for action - highlight message font	143
Time for action - highlight tabs font	144

The error state	147
Error borders	148
Time for action - error message borders	148
Time for action - error tabs border	149
Error background	151
Time for action - error message background	152
Time for action - accordion error background	153
Error font	155
Time for action - error message font	155
Time for action - error button font	157
The disabled state	159
Disabled opacity	159
Time for action - increasing disabled opacity	159
Priorities	161
Time for action - button priorities	161
Summary	163
Chapter 7: Creating Theme Icons	165
What are theme icons?	165
Time for action - preparing the example	166
Icons in widgets	167
Time for action - default widget icons	167
Time for action - setting widget icons	170
Standalone icons	172
Time for action - displaying the current user	172
Time for action - identifying entities	173
Icon states	177
Time for action - default icons	177
Time for action - highlight icons	178
Time for action - error icons	179
Time for action - hover icons	180
Sprites	181
What are sprites?	182
Icon categories	182
Theme icon sets	183
Time for action - creating new icons	183
Summary	190
Chapter 8: Special Effects	191
Setting up a jQuery environment	191
Theme corners	193
The legacy approach	193

Table of Contents

The style approach	194
The jQuery UI approach	196
Time for action - setting all corners	196
Time for action - setting top corners	197
Time for action - setting individual corners	198
Time for action - increasing corner roundness	200
Time for action - complex radius values	202
Time for action - using percentages	204
Time for action - removing corners	206
Theme shadows	209
Time for action - applying shadows	209
Time for action - altering shadows	211
Time for action - changing perspectives	212
Theme overlays	213
Time for action - adjusting dialog overlays	214
Summary	216
Chapter 9: Theming Custom Widgets	217
What are custom widgets?	217
Widget JS	218
Widget CSS	218
Implementing widgets	220
The widget	220
Time for action – basic markup	221
Goals of the widget	223
Time for action – basic JavaScript	223
Time for action – adding more behavior	227
Styling widgets	230
Time for action – widget CSS	230
Time for action – theme CSS	236
Time for action – finishing touches	238
Summary	244
Pop Quiz Answers	245
Index	247

Preface

Welcome to *jQuery UI Themes: Beginner's Guide*. This introductory text will get you started with developing your own themes for jQuery UI applications. Starting with the general concepts of user interface themes, this book walks the reader through everything from the most basic ideas in jQuery UI to more advanced topics such as icons and themes for custom widgets.

What this book covers

Chapter 1, Themeable User Interfaces, talks about themes in general and why they're important for user interfaces. We'll also address some introductory jQuery UI theme concepts.

Chapter 2, Using Themes, explains how to apply jQuery UI themes by example. Here, we'll walk through some introductory examples that use jQuery UI themes.

Chapter 3, Using the ThemeRoller, introduces the ThemeRoller application. This chapter walks the reader through all available ThemeRoller settings and shows how to download and use your theme.

Chapter 4, Working with Widget Containers, explains how to theme widget containers. Containers are the basic building block for themes and this chapter explains a little theory and gives plenty of examples.

Chapter 5, Transforming Interaction States, talks about the different states jQuery UI widgets go through. These states have different visual appearances and you'll find numerous examples on how to customize them.

Chapter 6, Customizing Interaction Cues, gives examples of using interaction cues with jQuery UI. There is a little theory on what they are, as well as examples on how to theme them.

Chapter 7, Creating Theme Icons, talks about icon sets in jQuery UI themes. We cover how these icons are stored in the image files and how to add your own icons to a theme.

Chapter 8, Special Effects, talks about special effects in the jQuery UI theme framework – rounded corners, and shadows. You'll see several examples of how to apply these classes to widgets and how to customize them.

Chapter 9, Theming Custom Widgets, walks you through the process of constructing a custom widget. We then cover making the widget theme-ready.

What you need for this book

All you need to run the examples in this book are the following:

- ◆ An Internet connection (for downloading jQuery UI)
- ◆ A web browser
- ◆ A text editor
- ◆ GIMP image editor (free) or PhotoShop (this is required only for chapter 7)

Who this book is for

This book is intended for anyone interested in learning about jQuery UI themes.

Conventions

In this book, you will find several headings appearing frequently.

To give clear instructions of how to complete a procedure or task, we use:

Time for action – heading

Action 1

Action 2

Action 3

Instructions often need some extra explanation so that they make sense, so they are followed with:

What just happened?

This heading explains the working of tasks or instructions that you have just completed.

You will also find some other learning aids in the book, including:

Pop quiz – heading

These are short multiple choice questions intended to help you test your own understanding.

Have a go hero – heading

These set practical challenges and give you ideas for experimenting with what you have learned.

You will also find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "In `index.html` and `index.js`, we create three jQuery UI button widgets to show off our new icon."

A block of code is set as follows:


```
$(document).ready(function() {  
  
    $("#my_button").button();  
  
});
```


When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
#todo-title {  
    color: #222222;  
    padding-left: 5%;  
}  
  
#todo-list {  
    font-size: 0.8em;  
    color: #362b36;  
    list-style-type: square;  
}
```

```
h3, ul {  
  
    font-family: Lucida Grande, Lucida Sans, Arial, sans-serif;  
  
}
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "If the **Layers** dialog isn't open already, open it by selecting **Windows | Dockable Dialogs | Layers** as illustrated".

 Warnings or important notes appear in a box like this.

 Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title via the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on www.packtpub.com or e-mail suggest@packtpub.com.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code for this book

You can download the example code files for all Packt books you have purchased from your account at <http://www.PacktPub.com>. If you purchased this book elsewhere, you can visit <http://www.PacktPub.com/support> and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Themeable User Interfaces

Web developers use styles effectively to make web pages look professional. HTML is the structural backbone behind user interfaces viewed on the Web. We can do a lot of interesting things with the look and feel of the Web user experience. We can take an otherwise boring user interface and breathe some life into it.

Giving a web page a look and feel poses the same problem faced by any other software development discipline—how do we prevent ourselves from reinventing the wheel while maintaining usability? We build themeable user interfaces to overcome obstacles such as these.

In this chapter, we will learn the following:

- ◆ What are themes and widgets and how do they relate?
- ◆ How to create a basic widget and apply styles to it
- ◆ Applying basic theming concepts to CSS styles
- ◆ The basic ideas behind the jQuery UI framework

So let's get on with it.

What is a theme?

Themes represent consistency and familiarity in graphical user interfaces. That is to say, a user interface shouldn't contradict itself by looking vastly different in various contexts. The concept of themes applies to many things humans interact with. This includes both computer user interfaces, as well as those found in the real world.

Themes in the real world

All kinds of real-world things have a theme. Groups of related things share some similarity or another. This also means that they are consistent in some respect. A product line generally has a theme.

A car is a good example of a themed object. Have you ever seen a car in the distance and thought "that is unmistakably Ford"? You don't put a lot of thought into what makes it a Ford, it "just looks like one". This feeling of familiarity is true of different car types. Ford cars and Ford trucks still look like Ford.

This goes beyond the branding that the company logo on the product gives. There are subtleties that give us visual cues that this is a Ford or this is a Mercedes. When either of the companies comes up with a new car, you would instantly know where it came from.

These hidden subtleties that are constant throughout a product line are an example of a theme. Going back to the car example, it is usually the overall shape of the car that give it its distinctiveness. The small grooves in the body are usually consistent across different models from the manufacture.

The color of a product alone doesn't necessarily dictate the theme in the real world. Seeing the same Subaru wagon in red doesn't hinder our ability to identify the make.

Desktop themes

Your desktop user interface is also themeable. Desktops are made up of windows and other elements within them. The look and feel these elements provide vary with different operating systems. But the function, or the purpose, of say, a button, is universal.

We can change the look and feel of most popular desktop environments by simply changing a preference setting. We can select from a list of available themes that ship with the operating system or we can download a third-party theme someone has created.

We can design desktop themes because there is an API to do this. If a theme author wants to change the border color of all desktop windows, they can do this without changing what a window does. The window looks different but doesn't behave differently.

Themes on the Web

User interfaces built for Web applications don't share the same flexibility as those built for the desktop. There is really no limitation that says we can't build themes that are portable enough to work with more than one application in a web browser. The underlying functionality that gives the desktop applications their look and feel isn't very different from that of a web application.

We can drastically alter the look and feel of a web application with **styles**. Style can be applied to a single HTML element or a range of elements. We have the flexibility to pick and choose which elements we change.

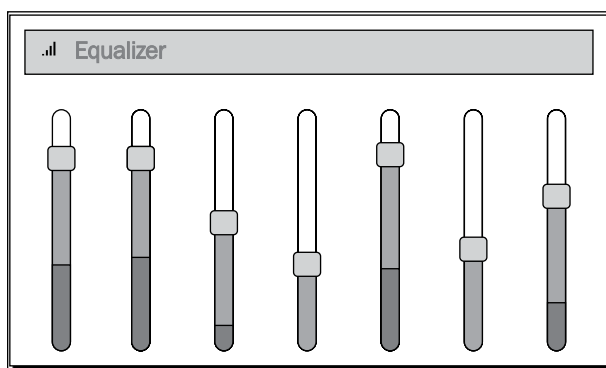
With HTML, there are a variety of ways to achieve the same visual result. For instance, creating portable themes that work across applications. If there was a consistent structure we could apply style to, themes become much simpler to develop, and keep portable. As with the desktop environment, widgets are used in a web environment to make user interfaces themeable.

What are widgets?

Before we start creating our own widgets and styling them, let's take a moment to discuss what widgets actually are. A widget, in the user interface sense, is a portion of anything displayed on the screen. Widgets can be large and complex, or small and simplistic. They usually have at least some structural element that is defined by the developer. A block of text isn't a widget; the developer doesn't specify the shape of each character. Understanding what widgets are and how they relate to themes is important for building themeable user interfaces. For example, the date picker widget solves a common problem—allowing the user to select a date. Themes control how the date picker looks.

Widget structure

Widgets aren't just a single, opaque whole. jQuery UI widgets are structured as assemblies of parts. For instance, below is an equalizer widget. You can see it is made up of several components – the header, and individual sliders for controlling the widget:



So what are the individual parts that make up a widget? Is there a predefined set of fields that a user interface component must have in order to be considered a widget? There are no qualifying attributes that make something a widget. The fields that define a widget structure are up to the widget creator. They can have a single field and the widget does the rest, or, they can have dozens of fields that allow for a lot of flexibility and are harder to use.

Widgets can contain other widgets. These subordinate widgets can contain others, and so on. We can continue down this path, moving ever further down the ladder until we finally reach the lowest level HTML element. In practice, however, we don't want a complex widget structure that is difficult to style.

Widgets and themes

So how does the structure of a widget, or widgets in general for that matter, relate to themes? Applying a style to a standard HTML element is easy to do. However, in order to share themes, to make them portable, we need to share structure. Theme authors don't design themes for one specific application. They're intended to be used and reused with different applications. The same idea is true for widgets, jQuery UI widgets, Dojo widgets, and so on; they're all intended for reuse.

Portability is an important attribute of widgets. Applying styles to widgets, changing their look and feel, isn't all that different from styling plain old HTML elements. The difference is that the consistency with which the styles are applied to widgets remains the same across applications.

Styling user interfaces

Creating styles that change the look and feel of web application is the reason CSS exists. The many presentation properties that define a style are stored in a style sheet, separate from the structural elements of the user interface. We could directly apply styles to the element as a style attribute. By doing this, we lose all portability; the style is forever bound to the element.

There is more to defining the look and feel of your application when it comes to styling user interfaces. A big part of style creation is structuring your CSS in such a way that they aren't restrictive. Let's take a closer look at what is involved with styling user interfaces.

Time for action - creating a style

Now that we have a general idea of what a widget is and how they relate to themes, let's create one. We're going to create a todo list widget. This widget isn't going to be built for any particular application. We want to reuse it somewhere else, should the need arise. The requirements of this todo list widget are pretty straightforward. It will display the title of the list along with the list items.

1. Create a new `index.html` file with the following content and save it:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Todo List</title>
    <link href="style.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <div id="container">
      <h3 id="todo-title">TODO</h3>
      <ul id="todo-list">
        <li>Design user interface</li>
        <li>Implement form validation</li>
        <li>Deploy</li>
      </ul>
    </div>
  </body>
</html>
```

2. In the same directory which has the `index.html` file, create a new `style.css` file and save it:

```
#container {

  background: #f2f5f7;
  border: 3px solid #dddddd;
  width: 25%;
  padding: 3px;

}

#todo-title {

  font-family: Lucida Grande, Lucida Sans, Arial, sans-serif;
  color: #222222;
  padding-left: 5%;

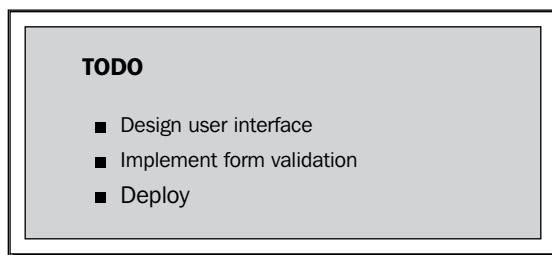
}

#todo-list {
```

```
font-family: Lucida Grande, Lucida Sans, Arial, sans-serif;
font-size: 0.8em;
color: #362b36;
list-style-type: square;

}
```

3. Open `index.html` in a web browser. In most operating systems, you should be able to right-click the file and see an option to open the file with the default browser. Alternatively, you can choose **File | Open** to select `index.html`:



What just happened?

Let's walk through the code and see exactly what we've just created.

The `index.html` file defines the structure of our todo list widget. The `title` element sets the page title to Todo List. The following `link` element imports the CSS file `style.css`. This is where the styles for our todo list widget are defined.

Inside the `body` tag is the actual todo list widget. The beginning of our widget is the `div` element with the `id` attribute `container`. Next is the todo list title. The widget title is inside the `h3` element with the `id` attribute `todo-title`. Finally, the main purpose of the widget is the list of todo items. The list is the `ul` element with the `id` attribute `todo-list`. Inside the list are all the list items, defined with `li` elements.

The `style.css` file creates the look and feel of our *todo list* widget. There are three styles here that are applied to our widget.

The `#container` style creates a light blue background using the `background` property. The border is displayed as solid gray color using the `border` property. The width is set to 25% of the page using the `width` property. We give the container's contents some additional space with the `padding` property.

The `#todo-title` style sets the font of the todo list title using the `font-family` property. The font color is set to a dark grey using the `color` property. We give a little space to the left-hand side of the title using the `padding-left` property.