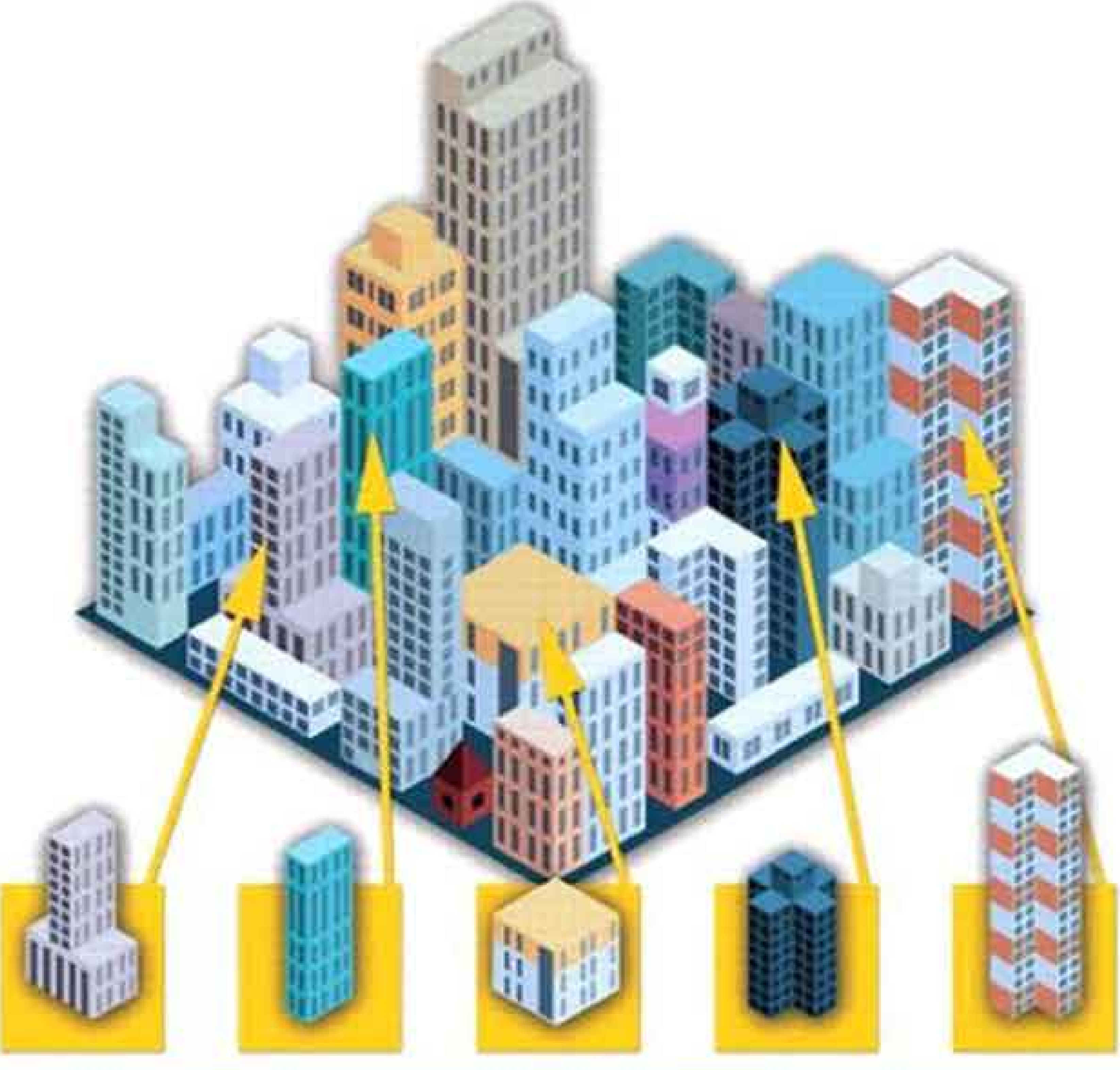
FUNDAMENTALS of WEB DEVELOPMENT



RANDY CONNOLLY • RICARDO HOAR

Fundamentals of Web Development

This page intentionally left blank

Fundamentals of Web Development

Randy Connolly
Mount Royal University, Calgary

Ricardo Hoar
Mount Royal University, Calgary

PEARSON

Boston Columbus Indianapolis New York San Francisco Upper Saddle River
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto
Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editorial Director: *Marcia Horton*
Acquisitions Editor: *Matt Goldstein*
Editorial Assistant: *Kelsey Loanes*
Program Manager: *Kayla Smith-Tarbox*
Marketing Coordinator: *Jon Bryant*
Managing Editor: *Scott Disanno*
Operations Supervisor: *Vincent Scelta*
Manufacturing Buyer: *Linda Sager*
Text Designer: *Jerilyn Bockorick, Cenveo® Publisher Services*
Cover Designer: *Marta Samsel*
Manager, Rights and Permissions: *Timothy Nicholls*
Text Permission Coordinator: *Jenell Forschler*
Cover Art: © *pilarts/Fotolia*
Media Project Manager: *Renata Butera*
Full-Service Project Management: *Hardik Popli, Cenveo Publisher Services*
Interior Printer/Bindery: *Courier/Kendallville*
Cover Printer: *Lehigh-Phoenix Color*

Credits and acknowledgments borrowed from other sources and reproduced, with permission, in this textbook appear on appropriate page within text.

Library of Congress Cataloging-in-Publication Data
Connolly, Randy, 1964-
Fundamentals of web development / Randy Connolly, Mount Royal University, Calgary ; Ricardo Hoar, Mount Royal University, Calgary.
pages cm
ISBN 978-0-13-340715-0 (alk. paper)
1. Web site development. I. Hoar, Ricardo. II. Title.
TK5105.888.C658 2014
006.7--dc23
2014003354

10 9 8 7 6 5 4 3 2 1—CK—15 14 13 12 11

PEARSON

ISBN 10: 0-13-340715-2
ISBN 13: 978-0-13-340715-0

To Janet, for your intelligence, support, beauty, and love.

Randy Connolly

Thanks be to you Joanne for the love and joy you bring to our family.

Ricardo Hoar

Brief Table of Contents

Chapter 1 How the Web Works 1

Chapter 2 Introduction to HTML 52

Chapter 3 Introduction to CSS 95

Chapter 4 HTML Tables and Forms 148

Chapter 5 Advanced CSS: Layout 184

Chapter 6 JavaScript: Client-Side Scripting 230

Chapter 7 Web Media 283

Chapter 8 Introduction to Server-Side Development with PHP 322

Chapter 9 PHP Arrays and Superglobals 364

Chapter 10 PHP Classes and Objects 402

Chapter 11 Working with Databases 436

Chapter 12 Error Handling and Validation 503

Chapter 13 Managing State 541

Chapter 14 Web Application Design 573

Chapter 15 Advanced JavaScript & jQuery 613

Chapter 16 Security 665

Chapter 17 XML Processing and Web Services 718

Chapter 18 Content Management Systems 781

Chapter 19 Web Server Administration 838

Chapter 20 Search Engines 881

Chapter 21 Social Network Integration 914

This page intentionally left blank

Table of Contents

Preface xxxiii

Acknowledgments xl

Chapter 1 How the Web Works 1

1.1 Definitions and History 2

A Short History of the Internet 2

The Birth of the Web 4

Web Applications in Comparison to Desktop Applications 6

Static Websites versus Dynamic Websites 8

Web 2.0 and Beyond 9

1.2 Internet Protocols 11

A Layered Architecture 12

Link Layer 12

Internet Layer 13

Transport Layer 15

Application Layer 16

1.3 The Client-Server Model 16

The Client 17

The Server 17

The Request-Response Loop 17

The Peer-to-Peer Alternative 18

Server Types 18

Real-World Server Installations 20

1.4 Where Is the Internet? 23

From the Computer to the Local Provider 24

From the Local Provider to the Ocean's Edge 26

Across the Oceans 29

1.5 Domain Name System 30

Name Levels 32

Name Registration 34

Address Resolution 34

1.6 Uniform Resource Locators 38

Protocol 38

Domain 39

Port 39

Path 39

Query String 39

Fragment 39

1.7 Hypertext Transfer Protocol 40

Headers 42

Request Methods 44

Response Codes 45

1.8 Web Servers 46

Operating Systems 47

Web Server Software 47

Database Software 48

Scripting Software 48

1.9 Chapter Summary 48

Key Terms 49

Review Questions 49

References 50

Chapter 2 Introduction to HTML 52

2.1 What Is HTML and Where Did It Come from? 53

XHTML 55

HTML5 57

2.2 HTML Syntax 59

Elements and Attributes 59

Nesting HTML Elements 60

2.3 Semantic Markup 62

2.4 Structure of HTML Documents 64

DOCTYPE 65

Head and Body 66

2.5 Quick Tour of HTML Elements 68

Headings 68

Paragraphs and Divisions 72

Links 72

URL Relative Referencing 74

Inline Text Elements 78

Images 78

Character Entities 79

Lists 80

2.6 HTML5 Semantic Structure Elements 81

Header and Footer 81

Heading Groups 84

Navigation 84

Articles and Sections 85

Figure and Figure Captions 87

Aside 89

2.7 Chapter Summary 89

Key Terms 89

Review Questions 90

Hands-On Practice 90

Chapter 3 Introduction to CSS 95

3.1 What Is CSS? 96

Benefits of CSS 96

CSS Versions 96

Browser Adoption 97

3.2 CSS Syntax 98

Selectors 99

Properties 99

Values 100

3.3 Location of Styles 103

Inline Styles 103

Embedded Style Sheet 104

External Style Sheet 104

3.4 Selectors 105

Element Selectors 106

Class Selectors 106

Id Selectors 107

Attribute Selectors 110

Pseudo-Element and Pseudo-Class Selectors 112

Contextual Selectors 114

3.5 The Cascade: How Styles Interact 116

Inheritance 116

Specificity 116

Location 119

3.6 The Box Model 122

Background 123

Borders 124

Margins and Padding 125

Box Dimensions 128

3.7 CSS Text Styling 134

Font Family 134

Font Sizes 136

Paragraph Properties 138

3.8 Chapter Summary 140

Key Terms 141

Review Questions 141

Hands-On Practice 142

References 147

Chapter 4 HTML Tables and Forms 148

4.1 Introducing Tables 149

Basic Table Structure 149

Spanning Rows and Columns 150

Additional Table Elements 151

Using Tables for Layout 152

4.2 Styling Tables 155

Table Borders 155

Boxes and Zebras 156

4.3 Introducing Forms 158

Form Structure 159

How Forms Work 160

Query Strings 161

The <form> Element 162

4.4 Form Control Elements 163

Text Input Controls 165

Choice Controls 167

Button Controls 169

Specialized Controls 171

Date and Time Controls 172

4.5 Table and Form Accessibility 174

Accessible Tables 175

Accessible Forms 176

4.6 Microformats 177

4.7 Chapter Summary 178

Key Terms 179

Review Questions 179

Hands-On Practice 180

Chapter 5 Advanced CSS: Layout 184

5.1 Normal Flow 185

5.2 Positioning Elements 188

Relative Positioning 188

Absolute Positioning 189

Z-Index 190

Fixed Position 191

5.3 Floating Elements 193

Floating within a Container 193

Floating Multiple Items Side by Side 195

Containing Floats 198

Overlaying and Hiding Elements 199

5.4 Constructing Multicolumn Layouts 203

Using Floats to Create Columns 204

Using Positioning to Create Columns 207

5.5 Approaches to CSS Layout 209

Fixed Layout 210

Liquid Layout 211

Other Layout Approaches 213

5.6 Responsive Design 214

Setting Viewports 215

Media Queries 218

5.7 CSS Frameworks 220

Grid Systems 220

CSS Preprocessors 222

5.8 Chapter Summary 225

Key Terms 225

Review Questions 225
Hands-On Practice 226

Chapter 6 JavaScript: Client-Side Scripting 230

6.1 What Is JavaScript and What Can It Do? 231
Client-Side Scripting 232
JavaScript's History and Uses 235
6.2 JavaScript Design Principles 240
Layers 241
Users without JavaScript 243
Graceful Degradation and Progressive Enhancement 247
6.3 Where Does JavaScript Go? 247
Inline JavaScript 249
Embedded JavaScript 249
External JavaScript 250
Advanced Inclusion of JavaScript 250
6.4 Syntax 251
Variables 252
Comparison Operators 252
Logical Operators 253
Conditionals 253
Loops 254
Functions 255
Errors Using Try and Catch 256
6.5 JavaScript Objects 257
Constructors 257
Properties 258
Objects Included in JavaScript 258
Window Object 261
6.6 The Document Object Model (DOM) 261
Nodes 262

Document Object 263
Element Node Object 265
Modifying a DOM Element 265
Additional Properties 268
6.7 JavaScript Events 268
Inline Event Handler Approach 268
Listener Approach 270
Event Object 271
Event Types 272
6.8 Forms 276
Validating Forms 276
Submitting Forms 278
6.9 Chapter Summary 278
Key Terms 278
Review Questions 279
Hands-On Practice 279
References 282
Chapter 7 Web Media 283
7.1 Digital Representations of Images 284
7.2 Color Models 288
RGB 288
CMYK 289
HSL 291
Opacity 292
Color Relationships 292
7.3 Image Concepts 296
Color Depth 296
Image Size 297
Display Resolution 301

7.4 File Formats 302
JPEG 302
GIF 303
PNG 308
SVG 308
Other Formats 310
7.5 Audio and Video 310
Media Concepts 310
Browser Video Support 312
Browser Audio Support 313
7.6 HTML5 Canvas 315
7.7 Chapter Summary 317
Key Terms 317
Review Questions 317
Hands-On Practice 318

Chapter 8 Introduction to Server-Side Development with PHP 322

8.1 What Is Server-Side Development? 323
Comparing Client and Server Scripts 323
Server-Side Script Resources 323
Comparing Server-Side Technologies 325
8.2 A Web Server's Responsibilities 328
Apache and Linux 329
Apache and PHP 330
PHP Internals 332
Installing Apache, PHP, and MySQL for Local Development 334
8.3 Quick Tour of PHP 336
PHP Tags 336
PHP Comments 337

Variables, Data Types, and Constants 339
Writing to Output 342
8.4 Program Control 346
if . . . else 346
switch . . . case 347
while and do . . . while 348
for 349
Alternate Syntax for Control Structures 349
Include Files 350
8.5 Functions 351
Function Syntax 352
Calling a Function 353
Parameters 353
Variable Scope within Functions 356
8.6 Chapter Summary 358
Key Terms 358
Review Questions 358
Hands-On Practice 359
References 363
Chapter 9 PHP Arrays and Superglobals 364
9.1 Arrays 365
Defining and Accessing an Array 365
Multidimensional Arrays 367
Iterating through an Array 367
Adding and Deleting Elements 369
Array Sorting 371
More Array Operations 372
Superglobal Arrays 373
9.2 $_GET and $_POST Superglobal Arrays 374
Determining If Any Data Sent 375

Accessing Form Array Data 378
Using Query Strings in Hyperlinks 379
Sanitizing Query Strings 380

9.3 $_SERVER Array 382
Server Information Keys 383
Request Header Information Keys 383

9.4 $_FILES Array 385
HTML Required for File Uploads 385
Handling the File Upload in PHP 386
Checking for Errors 388
File Size Restrictions 388
Limiting the Type of File Upload 390
Moving the File 391

9.5 Reading/Writing Files 392
Stream Access 392
In-Memory File Access 393

9.6 Chapter Summary 395
Key Terms 395
Review Questions 395
Hands-On Practice 396
References 401

Chapter 10 PHP Classes and Objects 402

10.1 Object-Oriented Overview 403
Terminology 403
The Unified Modeling Language 403
Differences between Server and Desktop Objects 404

10.2 Classes and Objects in PHP 407
Defining Classes 407
Instantiating Objects 408
Properties 408

Constructors 409
Methods 410
Visibility 412
Static Members 412
Class Constants 414
10.3 Object-Oriented Design 415
Data Encapsulation 415
Inheritance 420
Polymorphism 427
Object Interfaces 429
10.4 Chapter Summary 432
Key Terms 432
Review Questions 433
Hands-On Practice 433
References 435

Chapter 11 Working with Databases 436

11.1 Databases and Web Development 437
The Role of Databases in Web Development 437
Database Design 437
Database Options 443
11.2 SQL 445
SELECT Statement 445
INSERT, UPDATE, and DELETE Statements 447
Transactions 448
Data Definition Statements 453
Database Indexes and Efficiency 453
11.3 Database APIs 454
PHP MySQL APIs 455
Deciding on a Database API 455

11.4 Managing a MySQL Database 456
Command-Line Interface 456
phpMyAdmin 457
MySQL Workbench 459
11.5 Accessing MySQL in PHP 460
Connecting to a Database 460
Handling Connection Errors 462
Executing the Query 464
Processing the Query Results 470
Freeing Resources and Closing Connection 474
Using Transactions 475
11.6 Case Study Schemas 476
Art Database 477
Book CRM Database 477
Travel Photo Sharing Database 478
11.7 Sample Database Techniques 479
Display a List of Links 479
Search and Results Page 480
Editing a Record 484
Saving and Displaying Raw Files in the Database 492
11.8 Chapter Summary 495
Key Terms 496
Review Questions 496
Hands-On Practice 496
References 502

Chapter 12 Error Handling and Validation 503

12.1 What Are Errors and Exceptions? 504
Types of Errors 504
Exceptions 506

12.2 PHP Error Reporting 506
The error_reporting Setting 507
The display_errors Setting 507
The log_error Setting 508
12.3 PHP Error and Exception Handling 509
Procedural Error Handling 509
Object-Oriented Exception Handling 509
Custom Error and Exception Handlers 512
12.4 Regular Expressions 513
Regular Expression Syntax 513
Extended Example 516
12.5 Validating User Input 519
Types of Input Validation 519
Notifying the User 520
How to Reduce Validation Errors 521
12.6 Where to Perform Validation 524
Validation at the JavaScript Level 528
Validation at the PHP Level 531
12.7 Chapter Summary 536
Key Terms 536
Review Questions 537
Hands-On Practice 537
References 540
Chapter 13 Managing State 541
13.1 The Problem of State in Web Applications 542
13.2 Passing Information via Query Strings 544
13.3 Passing Information via the URL Path 546
URL Rewriting in Apache and Linux 546
13.4 Cookies 547
How Do Cookies Work? 548

Using Cookies 550
Persistent Cookie Best Practices 550

13.5 Serialization 552
Application of Serialization 554

13.6 Session State 554
How Does Session State Work? 557
Session Storage and Configuration 558

13.7 HTML5 Web Storage 561
Using Web Storage 561
Why Would We Use Web Storage? 563

13.8 Caching 563
Page Output Caching 565
Application Data Caching 565

13.9 Chapter Summary 567
Key Terms 567
Review Questions 568
Hands-On Practice 568
References 572

Chapter 14 Web Application Design 573

14.1 Real-World Web Software Design 574
Challenges in Designing Web Applications 574

14.2 Principle of Layering 575
What Is a Layer? 575
Consequences of Layering 577
Common Layering Schemes 579

14.3 Software Design Patterns in the Web Context 585
Adapter Pattern 585
Simple Factory Pattern 589
Template Method Pattern 591
Dependency Injection 594

14.4 Data and Domain Patterns 595

Table Data Gateway Pattern 596

Domain Model Pattern 597

Active Record Pattern 601

14.5 Presentation Patterns 604

Model-View-Controller (MVC) Pattern 604

Front Controller Pattern 607

14.6 Chapter Summary 608

Key Terms 608

Review Questions 608

Hands-On Practice 609

References 610

Chapter 15 Advanced JavaScript & jQuery 613

15.1 JavaScript Pseudo-Classes 614

Using Object Literals 614

Emulate Classes through Functions 615

Using Prototypes 617

15.2 jQuery Foundations 619

Including jQuery in Your Page 620

jQuery Selectors 621

jQuery Attributes 624

jQuery Listeners 628

Modifying the DOM 629

15.3 AJAX 633

Making Asynchronous Requests 636

Complete Control over AJAX 642

Cross-Origin Resource Sharing (CORS) 643

15.4 Asynchronous File Transmission 644

Old iframe Workarounds 645

The FormData Interface 646
Appending Files to a POST 648
15.5 Animation 649
Animation Shortcuts 649
Raw Animation 651
15.6 Backbone MVC Frameworks 654
Getting Started with Backbone.js 655
Backbone Models 655
Collections 657
Views 657
15.7 Chapter Summary 660
Key Terms 660
Review Questions 660
Hands-On Practice 661
References 664

Chapter 16 Security 665

16.1 Security Principles 666
Information Security 666
Risk Assessment and Management 667
Security Policy 670
Business Continuity 670
Secure by Design 673
Social Engineering 675
16.2 Authentication 676
Authentication Factors 676
Single-Factor Authentication 677
Multifactor Authentication 677
Third-Party Authentication 678
Authorization 681

16.3 Cryptography 681

Substitution Ciphers 683

Public Key Cryptography 686

Digital Signatures 689

16.4 Hypertext Transfer Protocol Secure (HTTPS) 690

Secure Handshakes 690

Certificates and Authorities 691

16.5 Security Best Practices 694

Data Storage 694

Monitor Your Systems 698

Audit and Attack Thyself 700

16.6 Common Threat Vectors 701

SQL Injection 701

Cross-Site Scripting (XSS) 703

Insecure Direct Object Reference 707

Denial of Service 708

Security Misconfiguration 709

16.7 Chapter Summary 712

Key Terms 713

Review Questions 713

Hands-On Practice 714

References 716

Chapter 17 XML Processing and Web Services 718

17.1 XML Overview 719

Well-Formed XML 719

Valid XML 720

XSLT 723

XPath 725

17.2 XML Processing 727

XML Processing in JavaScript 727

XML Processing in PHP 729

17.3 JSON 734

Using JSON in JavaScript 734

Using JSON in PHP 736

17.4 Overview of Web Services 737

SOAP Services 738

REST Services 740

An Example Web Service 740

Identifying and Authenticating Service Requests 744

17.5 Consuming Web Services in PHP 745

Consuming an XML Web Service 746

Consuming a JSON Web Service 750

17.6 Creating Web Services 756

Creating an XML Web Service 757

Creating a JSON Web Service 764

17.7 Interacting Asynchronously with Web Services 767

Consuming Your Own Service 768

Using Google Maps 769

17.8 Chapter Summary 774

Key Terms 775

Review Questions 775

Hands-On Practice 775

References 780

Chapter 18 Content Management Systems 781

18.1 Managing Websites 782

Components of a Managed Website 782

18.2 Content Management Systems 784

Types of CMS 785

18.3 CMS Components 787

Post and Page Management 787

WYSIWYG Editors 789
Template Management 790
Menu Control 791
User Management and Roles 791
User Roles 792
Workflow and Version Control 794
Asset Management 796
Search 797
Upgrades and Updates 799
18.4 WordPress Technical Overview 800
Installation 800
File Structure 801
WordPress Nomenclature 803
Taxonomies 806
WordPress Template Hierarchy 807
18.5 Modifying Themes 809
Changing Themes in Dashboard 809
Creating a Child Theme (CSS Only) 810
Changing Theme Files 811
18.6 Customizing WordPress Templates 812
WordPress Loop 812
Core WordPress Classes 813
Template Tags 815
Creating a Page Template 817
Post Tags 819
18.7 Creating a Custom Post Type 820
Organization 821
Registering Your Post Type 822
Adding Post-Specific Fields 823
Saving Your Changes 823
Under the Hood 824
Displaying Our Post Type 826

18.8 Writing a Plugin 828
Getting Started 828
Hooks, Actions, and Filters 829
Activate Your Plugin 830
Output of the Plugin 830
Make It a Widget 831
18.9 Chapter Summary 832
Key Terms 833
Review Questions 833
Hands-On Practice 833
References 837
Chapter 19 Web Server Administration 838
19.1 Web Server–Hosting Options 839
Shared Hosting 839
Dedicated Hosting 842
Collocated Hosting 843
Cloud Hosting 844
19.2 Domain and Name Server Administration 845
Registering a Domain Name 846
Updating the Name Servers 848
DNS Record Types 849
Reverse DNS 851
19.3 Linux and Apache Configuration 851
Configuration 853
Daemons 853
Connection Management 855
Data Compression 857
Encryption and SSL 858
Managing File Ownership and Permissions 860

19.4 Apache Request and Response Management 861

Managing Multiple Domains on One Web Server 861

Handling Directory Requests 863

Responding to File Requests 864

URL Redirection 864

Managing Access with .htaccess 868

Server Caching 870

19.5 Web Monitoring and Analytics 872

Internal Monitoring 872

External Monitoring 874

Internal Analytics 874

Third-Party Analytics 875

Third-Party Support Tools 875

19.6 Chapter Summary 877

Key Terms 877

Review Questions 877

Hands-On Practice 878

References 880

Chapter 20 Search Engines 881

20.1 The History and Anatomy of Search Engines 882

Before Google 882

Search Engine Overview 883

20.2 Web Crawlers and Scrapers 885

Robots Exclusion Standard 887

Scrapers 888

20.3 Indexing and Reverse Indexing 889

20.4 PageRank and Result Order 890

20.5 White-Hat Search Engine Optimization 894

Title 894

Meta Tags 895

URLs 896
Site Design 898
Sitemaps 899
Anchor Text 900
Images 901
Content 901
20.6 Black-Hat SEO 902
Content Spamming 902
Link Spam 904
Other Spam Techniques 906
20.7 Chapter Summary 908
Key Terms 908
Review Questions 909
Hands-On Practice 909
References 913

Chapter 21 Social Network Integration 914

21.1 Social Networks 915
How Did We Get Here? 915
Common Characteristics 918
21.2 Social Network Integration 919
Basic Social Media Presence 920
Facebook's Social Plugins 921
Open Graph 926
Google's Plugins 928
Twitter's Widgets 930
Advanced Social Network Integration 933
21.3 Monetizing Your Site with Ads 934
Web Advertising 101 934
Web Advertising Economy 937

21.4 Marketing Campaigns 938

Email Marketing 939

Physical World Marketing 943

21.5 Working in Web Development 945

Types of Web Development Companies 945

Roles and Skills 946

21.6 Chapter Summary 948

Key Terms 948

Review Questions 948

Hands-On Practice 949

References 953

Index 954

Preface

Welcome to the *Fundamentals of Web Development*. This textbook is intended to cover the broad range of topics required for modern web development and is suitable for intermediate to upper-level computing students. A significant percentage of the material in this book has also been used by the authors to teach web development principles to first-year computing students and to non-computing students as well.

One of the difficulties that we faced when planning this book is that web development is taught in a wide variety of ways and to a diverse student audience. Some instructors teach a single course that focuses on server-side programming to third-year students; other instructors teach the full gamut of web development across two or more courses, while others might only teach web development indirectly in the context of a networking, HCI, or capstone project course. We have tried to create a textbook that supports learning outcomes in all of these teaching scenarios.

What Is Web Development?

Web development is a term that takes on different meanings depending on the audience and context. In practice, web development requires people with complementary but distinct expertise working together toward a single goal. Whereas a graphic designer might regard web development as the application of good graphic design strategies, a database administrator might regard it as a simple interface to an underlying database. Software engineers and programmers might regard web development as a classic software development task with phases and deliverables, where a systems administrator sees a system that has to be secured from attackers. With so many different classes of user and meanings for the term, it's no wonder that web development is often poorly understood. Too often, in an effort to fully cover one aspect of web development, the other principles are ignored altogether, leaving students without a sense of where their skills fit into the big picture.

A true grasp of web development requires an understanding of multiple perspectives. As you will see, the design and layout of a website are closely related to the code and the database. The quality of the graphics is related to the performance and configuration of the server, and the security of the system spans every aspect of development. All of these seemingly independent perspectives are interrelated and therefore a web developer (of any type) should have a foundational understanding of all aspects, even if they only possess expertise in a handful of areas.

Features of the Book

To help students master the fundamentals of web development, this book has the following features:

- **Covers both the concepts and the practice of the entire scope of web development.** Web development can be a difficult subject to teach because it involves covering a wide range of theoretical material that is technology independent as well as practical material that is very specific to a particular technology. This book comprehensively covers both the conceptual and practical side of the entire gamut of the web development world.
- **Focused on the web development reality of today's world and in anticipation of future trends.** The world of web development has changed remarkably in the past decade. For instance, fewer and fewer sites are being created from scratch; instead, a great deal of current web development makes use of existing sophisticated frameworks and environments such as jQuery, WordPress, HTML5, and Facebook. We believe it is important to integrate this new world of web development into any web development textbook.
- **Sophisticated, realistic, and engaging case studies.** Rather than using simplistic "Hello World" style web projects, this book makes extensive use of three case studies: an art store, a travel photo sharing community, and a customer relations management system. For all the case studies, supporting material such as the business cases, use cases, design documentation, visual design, images, and databases are included. We have found that students are more enthusiastic and thus work significantly harder with attractive and realistic cases.
- **Comprehensive coverage of a modern Internet development platform.** In order to create any kind of realistic Internet application, readers require detailed knowledge of and practice with a single specific Internet development platform. This book covers HTML5, CSS3, JavaScript, and the LAMP stack (that is, Linux, Apache, MySQL, and PHP). Other important technologies covered include jQuery, XML, WordPress, Bootstrap, and a variety of third-party APIs that include Facebook, Twitter, and Google and Bing Maps.
- **Content presentation suitable for visually oriented learners.** As long-time instructors, the authors are well aware that today's students are often extremely reluctant to read long blocks of text. As a result, we have tried to

make the content visually pleasing and to explain complicated ideas not only through text but also through diagrams.

- **Content that is the result of over twenty years of classroom experience** (in college, university, and adult continuing education settings) teaching web development. The book's content also reflects the authors' deep experience engaging in web development work for a variety of international clients.
- **Tutorial-driven programming content available online**. Rather than using long programming listings to teach ideas and techniques, this book uses a combination of illustrations, short color-coded listings, and separate tutorial exercises. These step-by-step tutorials are not contained within the book, but are available online at **www.pearsonhighered.com/connolly-hoar**. Throughout the book you will find frequent links to these tutorial exercises.
- **Complete pedagogical features for the student**. Each chapter includes learning objectives, margin notes, links to step-by-step tutorials, advanced tips, keyword highlights, end-of-chapter review questions, and three different case study exercises.

Organization of the Book

The chapters in *Fundamentals of Web Development* can be organized into three large sections.

- **Foundational client-side knowledge (Chapters 1–7)**. These first chapters cover the foundational knowledge needed by any web developer. This includes how the web works (Chapter 1), HTML (Chapters 2 and 4), CSS (Chapters 3 and 5), JavaScript (Chapter 6), and web media (Chapter 7). Not every course would need to cover each of these chapters. Depending on the course, some instructors might skip Chapters 1, 5, 6, or 7.
- **Essential server-side development (Chapters 8–13)**. Despite the increasing importance of JavaScript-based development, learning server-side development is still the essential skill taught in most web development courses. The basics of PHP are covered in Chapters 8 and 9. Object-oriented PHP is covered in Chapter 10, and depending on the instructor, could be skipped (though PHP classes and objects are used in places in subsequent chapters). Database-driven web development is covered in Chapter 11, while state management and error handling are covered in Chapters 12 and 13.
- **Specialized topics (Chapters 14–21)**. Contemporary web development has become a very complex field, and different instructors will likely have different interest areas beyond the foundational topics. As such, our book provides specialized chapters that cover a variety of different interest areas.

Chapter 14 covers web application design for those interested more in software engineering and programming design. Chapter 15 includes advanced JavaScript and jQuery programming. Chapter 16 covers the vital topic of web security. Chapter 17 covers another programming topic: namely, consuming and creating web services. Chapter 18 covers the increasingly important topic of integrating with (and customizing) content management systems. The next two chapters address two important non-development topics: web server administration (Chapter 19) and search engines (Chapter 20). Finally, Chapter 21 covers another increasingly important topic: how to integrate a site into third-party social networks.

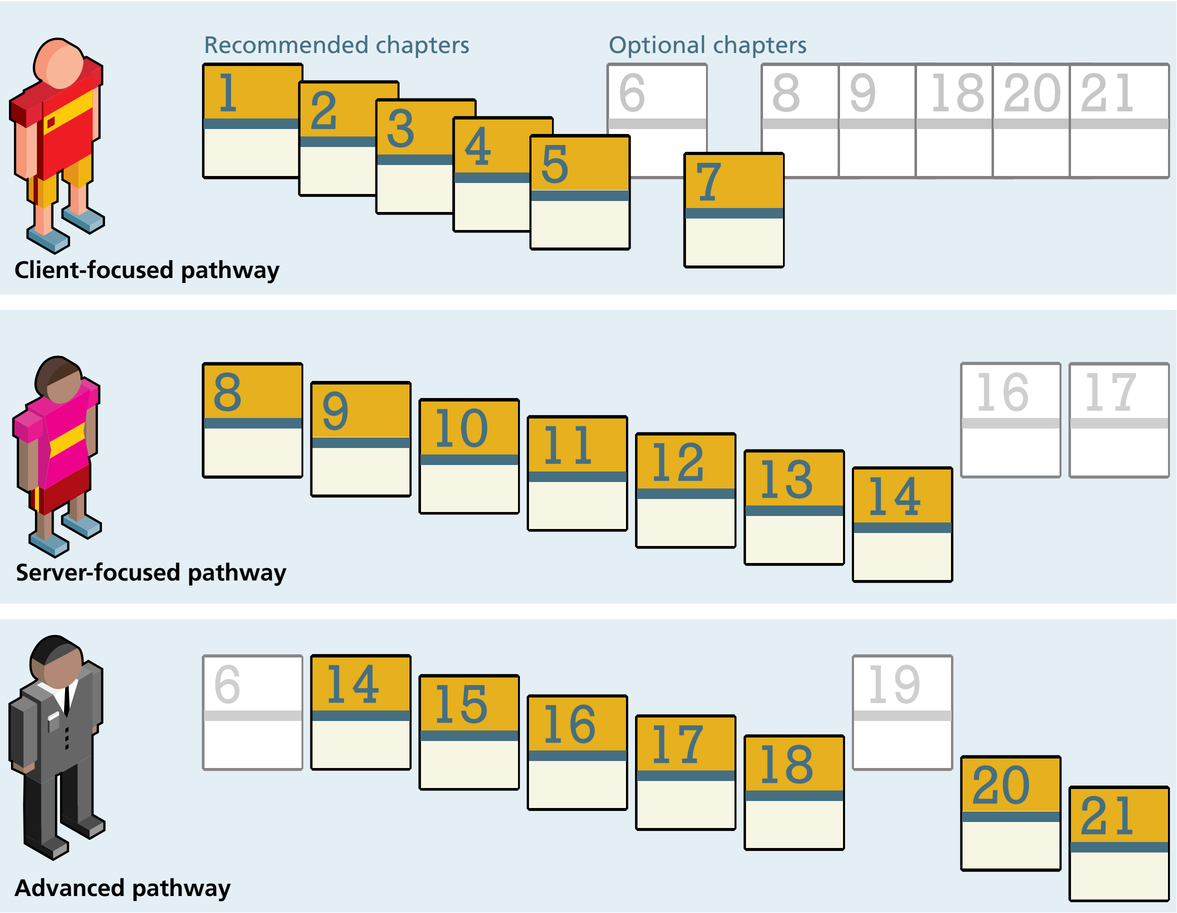
Pathways through this Book

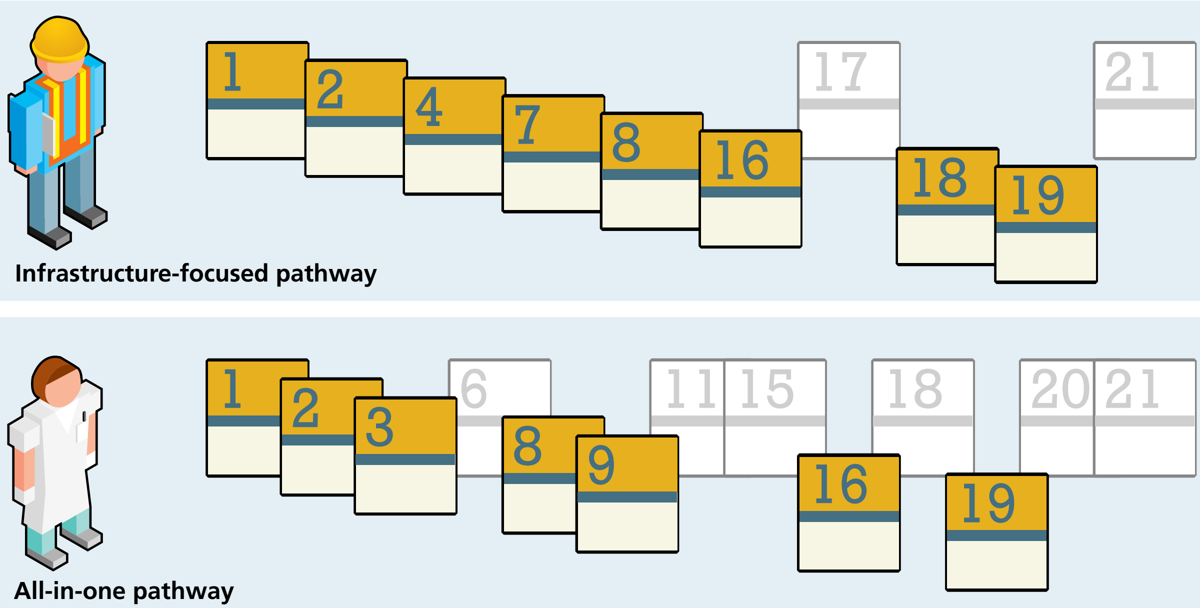
There are many approaches to teach web development and our book is intended to work with most of these approaches. It should be noted that this book has more material than can be plausibly covered in a single semester course. This is by design as it allows different instructors to chart their own unique way through the diverse topics that make up contemporary web development.

We do have some suggested pathways through the materials (though you are welcome to chart your own course), which you can see illustrated in the pathway diagrams.

- **All the web in a single course**. Many computing programs only have space for a single course on web development. This is typically an intermediate or upper-level course in which students will be expected to do a certain amount of learning on their own. In this case, we recommend covering Chapters 1, 2, 3, 4, 8, 9, 11, and 13. A semester-long course might also cover Chapters 6 and 16 as well.
- **Client-focused course for introductory students**. Some computing programs have a web course with minimal programming that may be open to non-major students or which acts as an introductory course to web development for major students. For such a course, we recommend covering Chapters 1, 2, 3, 4, 5, and 7. You can use Chapter 6 to introduce client-side scripting if desired. If some server-side web programming is going to be introduced, you can also cover Chapters 8 and 9. If no programming is going to be covered, you might consider adding some parts of Chapters 18, 20, and 21.
- **Server-focused course for intermediate students**. If students have already taken a client-focused course (or you want the students to learn the client content quickly on their own), then Chapters 8–14 and perhaps Chapters 16 and 17 would provide the students with a very solid foundation in server-side development.

- **Advanced web development course.** Some programs offer a web development course for upper-level students in which it is assumed that the students already know the foundational topics and are also experienced with the basics of server-side development. Such courses probably have the widest range of possible topics. One example of such a course that we have taught covers the content in Chapters 6 14–18, and 20–21.
- **Infrastructure-focused course.** In some computing programs the emphasis is less on the particulars of web programming and more on integrating web technologies into the overall computing infrastructure within an organization. Such a course might cover Chapters 1, 2, 4, 7, 8, 16, 18, 19, and part of Chapters 17 and 21.





For the Instructor

Web development courses have been called "unteachable" and indeed teaching web development has many challenges. We believe that using our book will make teaching web development significantly less challenging.

The following instructor resources are available at www.pearsonhighered.com/irc:

- Attractive and comprehensive PowerPoint presentations (one for each chapter).
- Images and databases for all the case studies.
- Solutions to end-of-chapter exercises and to tutorial exercises.

Why This Book?

The ACM computing curricula for computer science, information systems, information technology, and computing engineering all recommend at least a single course devoted to web development. As a consequence, almost every post-secondary computing program offers at least one course on web development.

Despite this universality, we could not find a suitable textbook for these courses that addressed both the theoretical underpinnings of the web together with modern web development practices. Complaints about this lack of breadth and depth have been well documented in published accounts in the computing education research literature. Although there are a number of introductory textbooks devoted to HTML and CSS, and, of course, an incredibly large number of trade books focused on specific web technologies, many of these are largely unsuitable for computing major students. Rather than illustrating how to create simple pages using HTML and JavaScript with very basic server-side capabilities, we believed that instructors increasingly need a textbook that guides students through the development of realistic, enterprise-quality web applications using contemporary Internet development platforms and frameworks.

This book is intended to fill this need. It covers the required ACM web development topics in a modern manner that is closely aligned with contemporary best practices in the real world of web development. It is based on our experience teaching a variety of different web development courses since 1997, our working professionally in the web development industry, our research in published accounts in the computing education literature, and in our corresponding with colleagues across the world. We hope that you find that this book does indeed satisfy your requirements for a web development textbook!

Acknowledgments

A book of this scale and scope incurs many debts of gratitude. We are first and foremost exceptionally grateful to Matt Goldstein, the Acquisitions Editor at Pearson, who championed the book and guided the overall process of bringing the book to market. Joan Murray and Shannon Bailey from Pearson played crucial roles in getting the initial prospectus considered. Kayla Smith-Tarbox was the Program Manager and ably handled the very tricky job of coordinating between the writers and the production team. Scott Disanno and Jenah Blitz-Soehr at Pearson also contributed in the early stages. We would like to thank Hardik Popli and his team at Cenveo Publisher Services for the work they did on the post-production side. We would also like to thank Margaret Berson, proofreader, who made sure that the words and illustrations actually work to tell a story that makes sense.

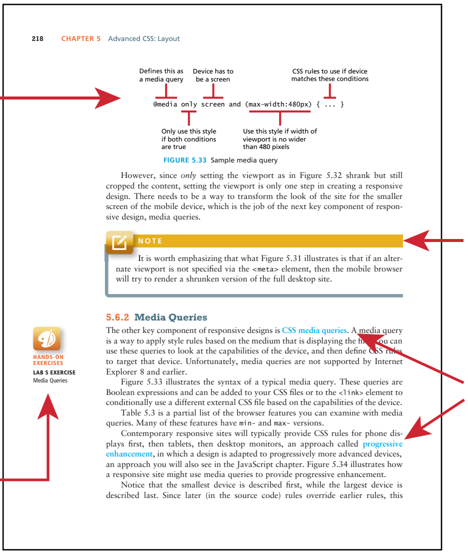
Reviewers help ensure that a textbook reflects more than just the authors' perspective. We were truly blessed in having two extraordinary reviewers: Jordan Pratt of Mount Royal University and Jamel Schiller of University of Wisconsin, Green Bay, who carefully examined every single chapter.

There are many others who helped guide our thinking, provided suggestions, or made our administrative and teaching duties somewhat less onerous. While we cannot thank everyone, we are grateful to Mount Royal University for granting a semester break for one of the authors, Peter Alston (now at the University of Liverpool) and his colleagues at Edge Hill University for hosting one of the authors for an important week early in the book's composition, and Amber Settle of De Paul University, who provided invaluable feedback on an early paper in which the rationale for the textbook was first hatched. Our long-time colleagues Paul Pospisil and Charles Hepler provided very helpful diversions from web development, which were always appreciated. And of course we would like to acknowledge all our students who have improved our insight and who acted as non-voluntary guinea pigs in the evolution of our thinking on teaching web development.

From its earliest inception in May of 2012 all the way to its conclusion in the early months of 2014, Dr. Janet Miller provided incredible and overwhelming encouragement, understanding, and feedback for which Randy Connolly will be always grateful. Joanne Hoar, an M.Sc. in computer science, made this book possible for Ricardo Hoar with continuous emotional support and professional feedback, all while maintaining a stable household for their three children under the age of 4 (and looking beautiful the whole time). Finally, we want to thank our children, Alexander Connolly, Benjamin Connolly, Archimedes Hoar, Curia Hoar, and Hypatia Hoar, who saw less of their fathers during this time but were always on our minds.

Visual Walkthrough

Hundreds of color-coded illustrations clarify key concepts.



218 CHAPTER 5 Advanced CSS: Layout

Defines this as a media query

Device has to be a screen

CSS rules to use if device matches these conditions

```
@media only screen and (max-width:480px) { ... }
```

Only use this style if both conditions are true

Use this style if width of viewport is no wider than 480 pixels

FIGURE 5.33 Sample media query

However, since *only* setting the viewport as in Figure 5.32 shrank but still cropped the content, setting the viewport is only one step in creating a responsive design. There needs to be a way to transform the look of the site for the smaller screen of the mobile device, which is the job of the next key component of responsive design, media queries.

NOTE

It is worth emphasizing that what Figure 5.31 illustrates is that if an alternate viewport is not specified via the <meta> element, then the mobile browser will try to render a shrunken version of the full desktop site.

5.6.2 Media Queries

HANDS-ON EXERCISES

LAB 5 EXERCISE

Media Queries

The other key component of responsive designs is CSS media queries. A media query is a way to apply style rules based on the medium that is displaying the file. You can use these queries to look at the capabilities of the device, and then define CSS rules to target that device. Unfortunately, media queries are not supported by Internet Explorer 8 and earlier.

Figure 5.33 illustrates the syntax of a typical media query. These queries are Boolean expressions and can be added to your CSS files or to the <link> element to conditionally use a different external CSS file based on the capabilities of the device.

Table 5.3 is a partial list of the browser features you can examine with media queries. Many of these features have min- and max- versions.

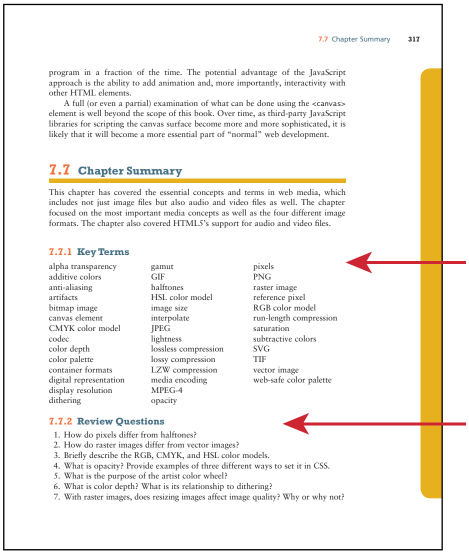
Contemporary responsive sites will typically provide CSS rules for phone displays first, then tablets, then desktop monitors, an approach called progressive enhancement, in which a design is adapted to progressively more advanced devices, an approach you will also see in the JavaScript chapter. Figure 5.34 illustrates how a responsive site might use media queries to provide progressive enhancement.

Notice that the smallest device is described first, while the largest device is described last. Since later (in the source code) rules override earlier rules, this

Security, Pro Tip, and Note boxes emphasize important concepts and practical advice.

Key terms are highlighted in consistent color.

Separate hands-on exercises (available online) give readers opportunity to practically apply concepts and techniques covered in the text.



7.7 Chapter Summary 317

program in a fraction of the time. The potential advantage of the JavaScript approach is the ability to add animation and, more importantly, interactivity with other HTML elements.

A full (or even a partial) examination of what can be done using the <canvas> element is well beyond the scope of this book. Over time, as third-party JavaScript libraries for scripting the canvas surface become more and more sophisticated, it is likely that it will become a more essential part of "normal" web development.

7.7 Chapter Summary

This chapter has covered the essential concepts and terms in web media, which includes not just image files but also audio and video files as well. The chapter focused on the most important media concepts as well as the four different image formats. The chapter also covered HTML5's support for audio and video files.

7.7.1 Key Terms

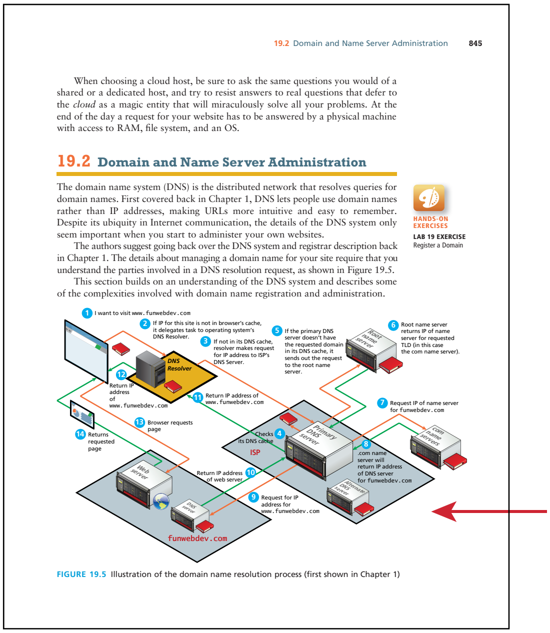
alpha transparency
additive colors
anti-aliasing
artifacts
bitmap image
canvas element
CMYK color model
codec
color depth
color palette
container formats
digital representation
display resolution
dithering
gamut
GIF
halftones
HSL color model
image size
interpolate
JPEG
lightness
lossless compression
lossy compression
LZW compression
media encoding
MPEG-4
opacity
pixels
PNG
raster image
reference pixel
RGB color model
run-length compression
saturation
subtractive colors
SVG
TIF
vector image
web-safe color palette

7.7.2 Review Questions

1. How do pixels differ from halftones?
2. How do raster images differ from vector images?
3. Briefly describe the RGB, CMYK, and HSL color models.
4. What is opacity? Provide examples of three different ways to set it in CSS.
5. What is the purpose of the artist color wheel?
6. What is color depth? What is its relationship to dithering?
7. With raster images, does resizing images affect image quality? Why or why not?

Key terms appear again at end of chapter.

Review questions at end of chapter provide opportunity for self-testing.



19.2 Domain and Name Server Administration 845

When choosing a cloud host, be sure to ask the same questions you would of a shared or a dedicated host, and try to resist answers to real questions that defer to the *cloud* as a magic entity that will miraculously solve all your problems. At the end of the day a request for your website has to be answered by a physical machine with access to RAM, file system, and an OS.

19.2 Domain and Name Server Administration

The domain name system (DNS) is the distributed network that resolves queries for domain names. First covered back in Chapter 1, DNS lets people use domain names rather than IP addresses, making URLs more intuitive and easy to remember. Despite its ubiquity in Internet communication, the details of the DNS system only seem important when you start to administer your own websites.

The authors suggest going back over the DNS system and registrar description back in Chapter 1. The details about managing a domain name for your site require that you understand the parties involved in a DNS resolution request, as shown in Figure 19.5.

This section builds on an understanding of the DNS system and describes some of the complexities involved with domain name registration and administration.

HANDS-ON EXERCISES

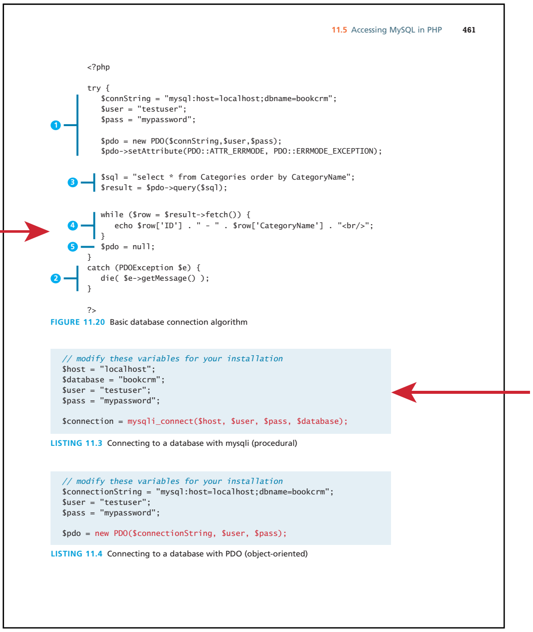
LAB 19 EXERCISE
Register a Domain

1 I want to visit www.funwebdev.com
2 If IP for this site is not in browser's cache, it delegates task to operating system's DNS Resolver.
3 If not in its DNS cache, resolver makes request for IP address to ISP's DNS Server.
4 Checks its DNS cache
5 If the primary DNS server doesn't have the requested domain in its DNS cache, it sends out the request to the root name server.
6 Root name server returns IP of name server for requested TLD (in this case the com name server).
7 Request IP of name server for funwebdev.com
8 .com name server will return IP address of DNS server for funwebdev.com
9 Request for IP address for www.funwebdev.com
10 Return IP address of web server
11 Return IP address of www.funwebdev.com
12 Return IP address of www.funwebdev.com
13 Browser requests page
14 Returns requested page
DNS Resolver
Root name server
Primary DNS server
ISP
com name servers
Alternate DNS server
Web server
DNS server
funwebdev.com

FIGURE 19.5 Illustration of the domain name resolution process (first shown in Chapter 1)

Illustrations help explain especially complicated processes.

Important algorithms are illustrated visually to help clarify understanding.



11.5 Accessing MySQL in PHP 461

```
   <?php

   try {
      $connString = "mysql:host=localhost;dbname=bookcrm";
      $user = "testuser";
1     $pass = "mypassword";

      $pdo = new PDO($connString,$user,$pass);
      $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

3     $sql = "select * from Categories order by CategoryName";
      $result = $pdo->query($sql);

      while ($row = $result->fetch()) {
4        echo $row['ID'] . " - " . $row['CategoryName'] . "<br/>";
      }
5     $pdo = null;
   }
   catch (PDOException $e) {
2     die( $e->getMessage() );
   }

   ?>
```

FIGURE 11.20 Basic database connection algorithm

```
// modify these variables for your installation
$host = "localhost";
$database = "bookcrm";
$user = "testuser";
$pass = "mypassword";

$connection = mysqli_connect($host, $user, $pass, $database);
```

LISTING 11.3 Connecting to a database with mysqli (procedural)

```
// modify these variables for your installation
$connectionString = "mysql:host=localhost;dbname=bookcrm";
$user = "testuser";
$pass = "mypassword";

$pdo = new PDO($connectionString, $user, $pass);
```

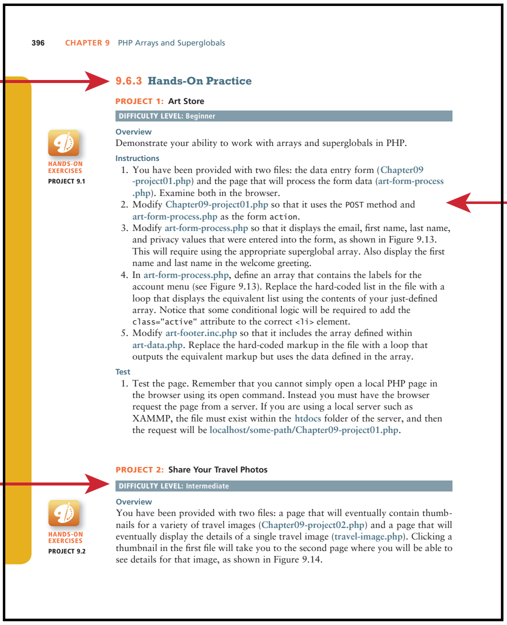
LISTING 11.4 Connecting to a database with PDO (object-oriented)

Color-coded source code listings emphasize important elements and visually separate comments from the code.

Each chapter ends with three case study exercises that allow the reader to practice the material covered in the chapter within a realistic context.

Exercises contain step-by-step instructions of varying difficulty.

Exercises increase in complexity and can be assigned separately by the instructor.



396 CHAPTER 9 PHP Arrays and Superglobals

9.6.3 Hands-On Practice

PROJECT 1: Art Store

DIFFICULTY LEVEL: Beginner

HANDS-ON EXERCISES
PROJECT 9.1

Overview

Demonstrate your ability to work with arrays and superglobals in PHP.

Instructions

1. You have been provided with two files: the data entry form (Chapter09-project01.php) and the page that will process the form data (art-form-process.php). Examine both in the browser.
2. Modify Chapter09-project01.php so that it uses the POST method and art-form-process.php as the form action.
3. Modify art-form-process.php so that it displays the email, first name, last name, and privacy values that were entered into the form, as shown in Figure 9.13. This will require using the appropriate superglobal array. Also display the first name and last name in the welcome greeting.
4. In art-form-process.php, define an array that contains the labels for the account menu (see Figure 9.13). Replace the hard-coded list in the file with a loop that displays the equivalent list using the contents of your just-defined array. Notice that some conditional logic will be required to add the class="active" attribute to the correct <li> element.
5. Modify art-footer.inc.php so that it includes the array defined within art-data.php. Replace the hard-coded markup in the file with a loop that outputs the equivalent markup but uses the data defined in the array.

Test

1. Test the page. Remember that you cannot simply open a local PHP page in the browser using its open command. Instead you must have the browser request the page from a server. If you are using a local server such as XAMPP, the file must exist within the htdocs folder of the server, and then the request will be localhost/some-path/Chapter09-project01.php.

PROJECT 2: Share Your Travel Photos

DIFFICULTY LEVEL: Intermediate

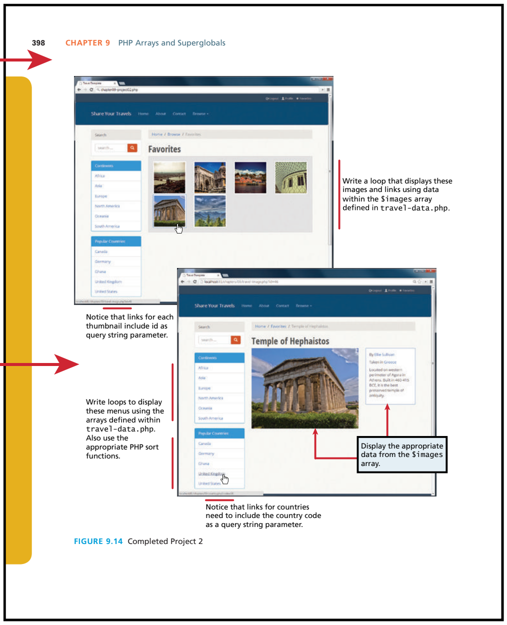
HANDS-ON EXERCISES
PROJECT 9.2

Overview

You have been provided with two files: a page that will eventually contain thumbnails for a variety of travel images (Chapter09-project02.php) and a page that will eventually display the details of a single travel image (travel-image.php). Clicking a thumbnail in the first file will take you to the second page where you will be able to see details for that image, as shown in Figure 9.14.

Attractive and realistic case studies help engage the readers' interest.

All images, pages, classes, databases, and other material for each of the case studies are available for download.



398 CHAPTER 9 PHP Arrays and Superglobals

Write a loop that displays these images and links using data within the $images array defined in travel-data.php.

Notice that links for each thumbnail include id as query string parameter.

Write loops to display these menus using the arrays defined within travel-data.php. Also use the appropriate PHP sort functions.

Display the appropriate data from the $images array.

Notice that links for countries need to include the country code as a query string parameter.

FIGURE 9.14 Completed Project 2

This page intentionally left blank

How the Web Works

1

CHAPTER OBJECTIVES

In this chapter you will learn . . .

- The history of the Internet and World Wide Web
- Fundamental concepts and protocols that support the Internet
- About the hardware and software that supports the Internet
- How a web page is actually retrieved and interpreted

This chapter introduces the World Wide Web (WWW). The WWW relies on a number of systems, protocols, and technologies all working together in unison. Before learning about HTML markup, CSS styling, JavaScript, and PHP programming, you must understand how the Internet makes web applications possible. This chapter begins with a brief history of the Internet and provides an overview of key Internet and WWW technologies applicable to the web developer. To truly understand these concepts in depth, one would normally take courses in computer science or information technology (IT) covering networking principles. If you find some of these topics too in-depth or advanced, you may decide to skip over some of the details here and return to them later.

1.1 Definitions and History

The World Wide Web (WWW or simply the Web) is certainly what most people think of when they see the word “Internet.” But the WWW is only a subset of the Internet, as illustrated in Figure 1.1.

1.1.1 A Short History of the Internet

The history of telecommunication and data transport is a long one. There is a strategic advantage in being able to send a message as quickly as possible (or at least, more quickly than your competition). The Internet is not alone in providing instantaneous digital communication. Earlier technologies like radio, telegraph, and the telephone provided the same speed of communication, albeit in an analog form.

Telephone networks in particular provide a good starting place to learn about modern digital communications. In the telephone networks of old, calls were routed through operators who physically connected caller and receiver by connecting a wire to a switchboard to complete a circuit. These operators were around in some areas for almost a century before being replaced with automatic mechanical switches, which did the same job: physically connect caller and receiver.

One of the weaknesses of having a physical connection is that you must establish a link and maintain a dedicated circuit for the duration of the call. This type of network connection is sometimes referred to as circuit switching and is shown in Figure 1.2.

The problem with circuit switching is that it can be difficult to have multiple conversations simultaneously (which a computer might want to do). It also requires more bandwidth since even the silences are transmitted (that is, unused capacity in the network is not being used efficiently).

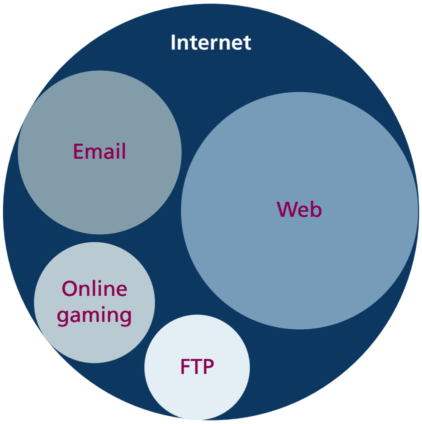


FIGURE 1.1 The web as a subset of the Internet

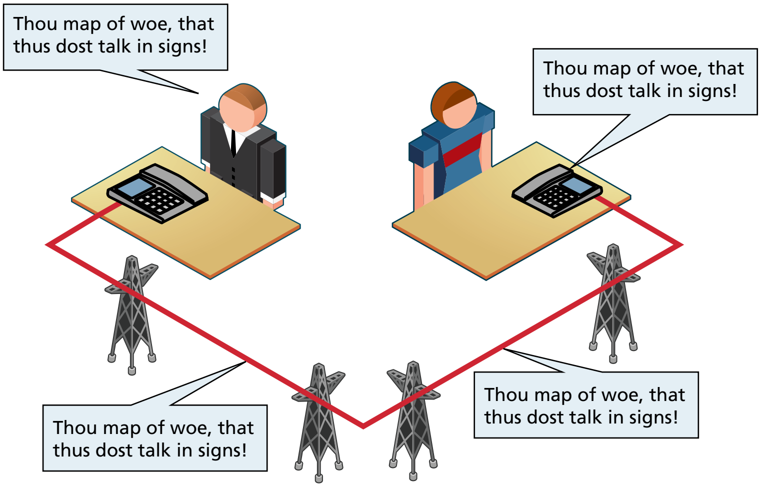

Thou map of woe, that thus dost talk in signs!
Thou map of woe, that thus dost talk in signs!
Thou map of woe, that thus dost talk in signs!
Thou map of woe, that thus dost talk in signs!

FIGURE 1.2 Telephone network as example of circuit switching

Bandwidth is a measurement of how much data can (maximally) be transmitted along an Internet connection. Normally measured in bits per second (bps), this measurement differs according to the type of Internet access technology you are using. A dial-up 56-Kbps modem has far less bandwidth than a 10-Gbps fiber optic connection.

In the 1960s, as researchers explored digital communications and began to construct the first networks, the research network ARPANET was created. ARPANET did not use circuit switching but instead used an alternative communications method called packet switching. A packet-switched network does not require a continuous connection. Instead it splits the messages into smaller chunks called packets and routes them to the appropriate place based on the destination address. The packets can take different routes to the destination, as shown in Figure 1.3. This may seem a more complicated and inefficient approach than circuit switching, but is in fact more robust (it is not reliant on a single pathway that may fail) and a more efficient use of network resources (since a circuit can communicate data from multiple connections).

This early ARPANET network was funded and controlled by the United States government, and was used exclusively for academic and scientific purposes. The early network started small with just a handful of connected university campuses and research institutions and companies in 1969 and grew to a few hundred by the early 1980s.

At the same time, alternative networks were created like X.25 in 1974, which allowed (and encouraged) business use. USENET, built in 1979, had fewer restrictions still, and as a result grew quickly to 550 hosts by 1981. Although there was growth in these various networks, the inability for them to communicate with each

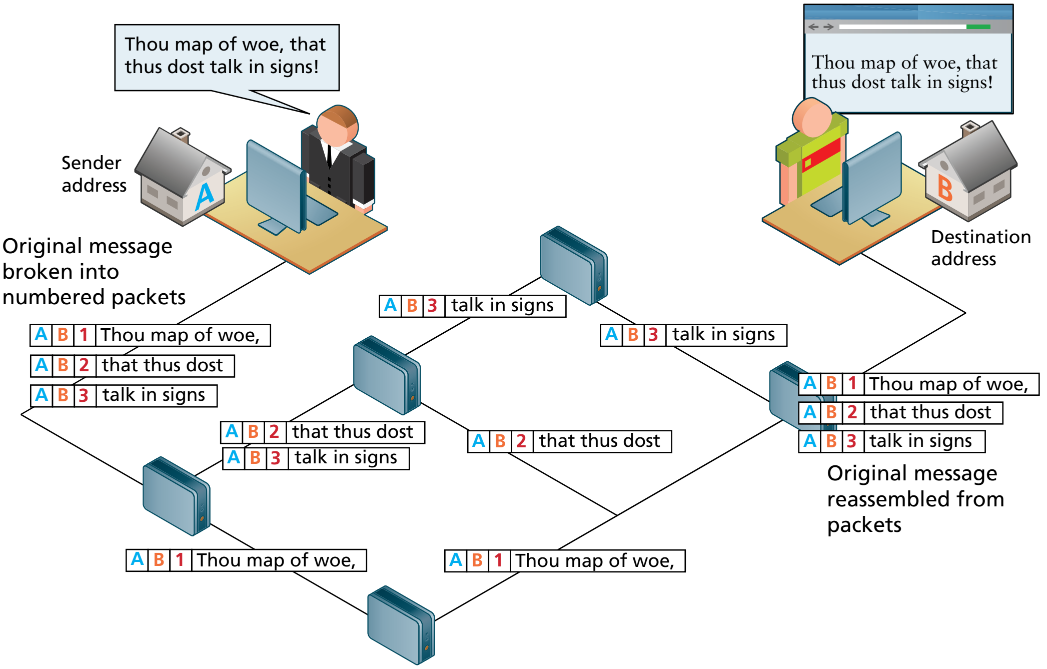


FIGURE 1.3 Internet network as example of packet switching

other was a real limitation. To promote the growth and unification of the disparate networks, a suite of protocols was invented to unify the networks. A protocol is the name given to a formal set of publicly available rules that manage data exchange between two points. Communications protocols allow any two computers to talk to one another, so long as they implement the protocol.

By 1981 protocols for the Internet were published and ready for use.[1,2] New networks built in the United States began to adopt the TCP/IP (Transmission Control Protocol/Internet Protocol) communication model (discussed in the next section), while older networks were transitioned over to it.

Any organization, private or public, could potentially connect to this new network so long as they adopted the TCP/IP protocol. On January 1, 1983, TCP/IP was adopted across all of ARPANET, marking the end of the research network that spawned the Internet.[3] Over the next two decades, TCP/IP networking was adopted across the globe.

1.1.2 The Birth of the Web

The next decade saw an explosion in the numbers of users, but the Internet of the late 1980s and the very early 1990s did not resemble the Internet we know today. During these early years, email and text-based systems were the extent of the Internet experience.

This transition from the old terminal and text-only Internet of the 1980s to the Internet of today is of course due to the invention and massive growth of the World Wide Web. This invention is usually attributed to the British Tim Berners-Lee (now Sir Tim Berners-Lee), who, along with the Belgian Robert Cailliau, published a proposal in 1990 for a hypertext system while both were working at CERN in Switzerland. Shortly thereafter Berners-Lee developed the main features of the web.[4]

This early web incorporated the following essential elements that are still the core features of the web today:

- A Uniform Resource Locator (URL) to uniquely identify a resource on the WWW.
- The Hypertext Transfer Protocol (HTTP) to describe how requests and responses operate.
- A software program (later called web server software) that can respond to HTTP requests.
- Hypertext Markup Language (HTML) to publish documents.
- A program (later called a browser) that can make HTTP requests from URLs and that can display the HTML it receives.

HTML will require several chapters to cover in this book. URLs and the HTTP are covered in this chapter. This chapter will also provide a little bit of insight into the nature of web server software; Chapter 20 will examine the inner workings of server software in more detail.

So while the essential outline of today's web was in place in the early 1990s, the web as we know it did not really begin until **Mosaic**, the first popular graphical browser application, was developed at the National Center for Supercomputing Applications at the University of Illinois Urbana-Champaign and released in early 1993 by Eric Bina and Marc Andreessen (who was a computer science undergraduate student at the time). Andreessen later moved to California and cofounded Netscape Communications, which released **Netscape Navigator** in late 1994. Navigator quickly became the principal web browser, a position it held until the end of the 1990s, when Microsoft's Internet Explorer (first released in 1995) became the market leader, a position it would hold for over a decade.

Also in late 1994, Berners-Lee helped found the **World Wide Web Consortium (W3C)**, which would soon become the international standards organization that would oversee the growth of the web. This growth was very much facilitated by the decision of CERN to not patent the work and ideas done by its employee and instead leave the web protocols and code-base royalty free.

To illustrate the growth of the Internet, Figure 1.4 graphs the count of hosts connected to the Internet from 1990 until 2010. You can see that the last decade in particular has seen an enormous growth, during which social networks, web

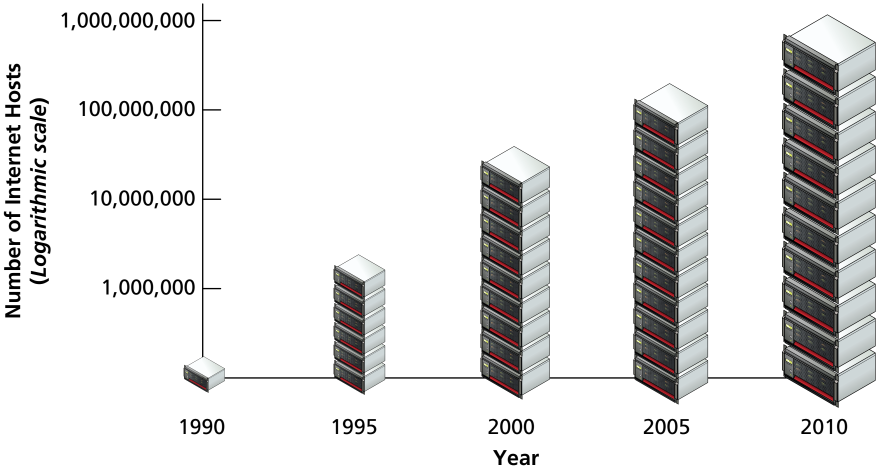


FIGURE 1.4 Growth in Internet hosts/servers based on data from the Internet Systems Consortium.[5]

services, asynchronous applications, the semantic web, and more have all been created (and will be described fully in due course in this textbook).

BACKGROUND

The Request for Comments (RFC) archive lists all of the Internet and WWW protocols, concepts, and standards. It started out as an unofficial repository for ARPANET information and eventually became the de facto official record. Even today new standards are published there.

1.1.3. Web Applications in Comparison to Desktop Applications

The user experience for a website is unlike the user experience for traditional desktop software. The location of data storage, limitations with the user interface, and limited access to operating system features are just some of the distinctions. However, as web applications have become more and more sophisticated, the differences in the user experience between desktop applications and web applications are becoming more and more blurred.

There are a variety of advantages and disadvantages to web-based applications in comparison to desktop applications. Some of the advantages of web applications include:

- Accessible from any Internet-enabled computer.
- Usable with different operating systems and browser applications.

- Easier to roll out program updates since only software on the server needs to be updated and not on every desktop in the organization.
- Centralized storage on the server means fewer security concerns about local storage (which is important for sensitive information such as health care data).

Unfortunately, in the world of IT, for every advantage, there is often a corresponding disadvantage; this is also true of web applications. Some of these disadvantages include:

- Requirement to have an active Internet connection (the Internet is not always available everywhere at all times).
- Security concerns about sensitive private data being transmitted over the Internet.
- Concerns over the storage, licensing, and use of uploaded data.
- Problems with certain websites on certain browsers not looking quite right.
- Restrictions on access to the operating system can prevent software and hardware from being installed or accessed (like Adobe Flash on iOS).

In addition, clients or their IT staff may have additional plugins added to their browsers, which provide added control over their browsing experience, but which might interfere with JavaScript, cookies, or advertisements. We will continually try to address these challenges throughout the book.

BACKGROUND

One of the more common terms you might encounter in web development is the term "intranet" (with an "a"), which refers to an Internet network that is local to an organization or business. Intranet resources are often private, meaning that only employees (or authorized external parties such as customers or suppliers) have access to those resources. Thus Internet (with an "e") is a broader term that encompasses both private (intranet) and public networked resources.

Intranets are typically protected from unauthorized external access via security features such as firewalls or private IP ranges, as shown in Figure 1.5. Because intranets are private, search engines such as Google have limited or no access to content within them.

Due to this private nature, it is difficult to accurately gauge, for instance, how many web pages exist within intranets, and what technologies are more common in them. Some especially expansive estimates guess that almost half of all web resources are hidden in private intranets.

Being aware of intranets is also important when one considers the job market and market usage of different web technologies. If one focuses just on the

(*continued*)

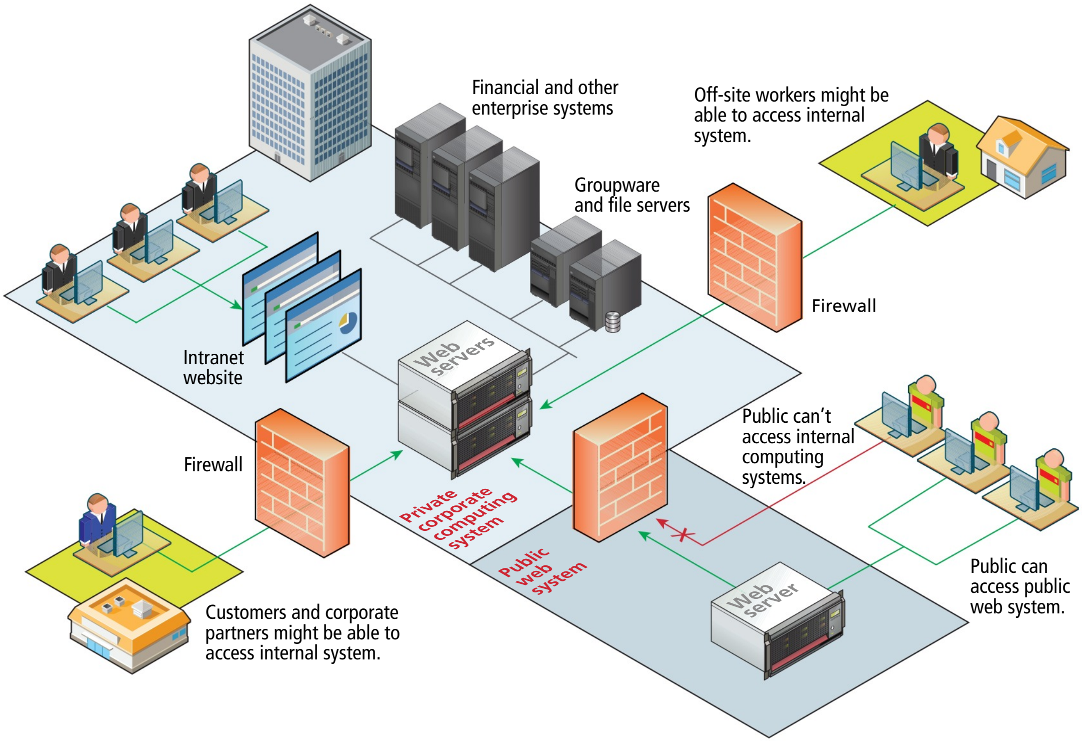

Financial and other enterprise systems
Off-site workers might be able to access internal system.
Groupware and file servers
Firewall
Intranet website
Web servers
Firewall
Public can't access internal computing systems.
Private corporate computing system
Public web system
Public can access public web system.
Web server
Customers and corporate partners might be able to access internal system.

FIGURE 1.5 Intranet versus Internet

public Internet, it will appear that PHP, MySQL, and WordPress are the most commonly used web development stack. But when one adds in the private world of corporate intranets, other technologies such as ASP.NET, JSP, SharePoint, Oracle, SAP, and IBM WebSphere are just as important.

1.1.4 Static Websites versus Dynamic Websites

In the earliest days of the web, a webmaster (the term popular in the 1990s for the person who was responsible for creating and supporting a website) would publish web pages and periodically update them. Users could read the pages but could not provide feedback. The early days of the web included many encyclopedic, collection-style sites with lots of content to read (and animated icons to watch).

In those early days, the skills needed to create a website were pretty basic: one needed knowledge of the HTML and perhaps familiarity with editing and creating images. This type of website is commonly referred to as a static website, in that it consists

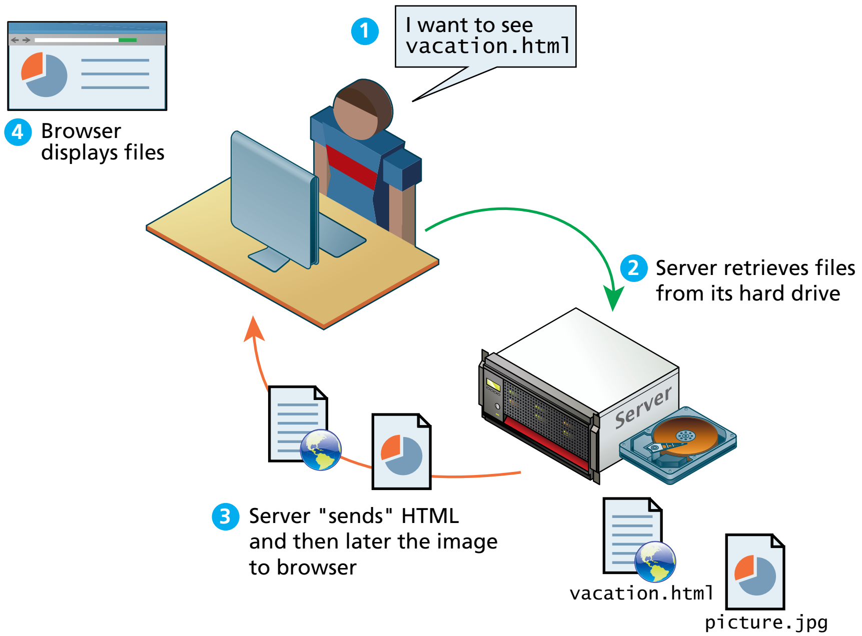

1 I want to see vacation.html
2 Server retrieves files from its hard drive
3 Server "sends" HTML and then later the image to browser
4 Browser displays files
Server
vacation.html
picture.jpg

FIGURE 1.6 Static website

only of HTML pages that look identical for all users at all times. Figure 1.6 illustrates a simplified representation of the interaction between a user and a static website.

Within a few years of the invention of the web, sites began to get more complicated as more and more sites began to use programs running on web servers to generate content dynamically. These server-based programs would read content from databases, interface with existing enterprise computer systems, communicate with financial institutions, and then output HTML that would be sent back to the users' browsers. This type of website is called here in this text a dynamic website because the page content is being created at run time by a program created by a programmer; this page content can vary from user to user. Figure 1.7 illustrates a very simplified representation of the interaction between a user and a dynamic website.

So while knowledge of HTML was still necessary for the creation of these dynamic websites, it became necessary to have programming knowledge as well. And by the late 1990s, other knowledge and skills were becoming necessary, such as CSS, usability, and security.

1.1.5 Web 2.0 and Beyond

In the mid-2000s, a new buzzword entered the computer lexicon: Web 2.0. This term had two meanings, one for users and one for developers. For the users, Web 2.0