# **Fundamental of Web Programming**

## **MDN Curriculum**

|  |  |
| --- | --- |
|  | **MDN Curriculum** |
|  | **Module-01: Getting Started** |
|  | **Soft Skills** |
| 01 | Constant Learning Mindset |
| 02 | Open to Embracing Failure |
| 03 | Effective Research |
| 04 | Collaboration and Teamwork |
| 05 | Succeeding in Job Interviews |
| 06 | Workflows and Processes |
| 07 | Relevant Contextual Information |
|  | **Environment Setup** |
| 08 | Computer Basics |
| 09 | File Systems |
| 10 | Browsing the Web |
| 11 | Command Line Basics |
| 12 | Code Editors |
|  | **MDN Curriculum** |
|  | **Module-02: Core** |
|  | **Web Standards** |
| 01 | How the Web Works |
| 02 | The HTML, CSS, and JavaScript Triangle |
| 03 | The Web Standards Model |
| 04 | How Browsers Load Webpages |
|  | **Semantic HTML** |
| 05 | Basic HTML Syntax |
| 06 | Good Document Structure |
| 07 | Lists |
| 08 | Advanced Text Techniques |
| 09 | Links |
| 10 | Media |
| 11 | Other Interactive Elements |
| 12 | HTML Tables |
| 13 | Debugging HTML |
|  | **CSS Fundamentals** |
| 14 | Basic CSS Syntax |
| 15 | Selectors |
| 16 | The Box Model |
| 17 | Handling Conflicts in CSS |
| 18 | Values and Units |
| 19 | Sizing |
| 20 | Backgrounds and Borders |
| 21 | Overflow |
| 22 | Styling Form Elements |
| 23 | Debugging CSS |
|  | **CSS Text Styling** |
| 24 | Text and Font Styling |
| 25 | Styling Lists and Links |
| 26 | Web Fonts |
|  | **CSS Layout** |
| 27 | CSS Layout Basics |
| 28 | Floats |
| 29 | Positioning |
| 30 | Modern Layout |
| 31 | Responsive Design |
|  | **JavaScript Fundamentals** |
| 32 | Variables |
| 33 | Math |
| 34 | Text |
| 35 | Arrays |
| 36 | Conditionals |
| 37 | Loops |
| 38 | Functions |
| 39 | JavaScript Object Basics |
| 40 | DOM Scripting |
| 41 | Events |
| 42 | Async JavaScript Basics |
| 43 | Network Requests with (fetch()) |
| 44 | Working with JSON |
| 45 | Libraries and Frameworks |
| 46 | Debugging JavaScript |
|  | **Accessibility** |
| 47 | Accessibility Basics |
| 48 | Accessible Styling |
| 49 | Accessible JavaScript |
| 50 | Assistive Technology |
| 51 | WAI-ARIA |
|  | **Design for Developers** |
| 52 | Basic Design Theory |
| 53 | User-Centered Design |
| 54 | Design Briefs |
| **55** | **Version Control** |
|  | **Module-03: Extensions** |
| **56** | **Transform and Animate CSS** |
| **57** | **Custom JS Objects** |
|  | **Web APIs** |
| 58 | Video and Audio APIs |
| 59 | Graphics / Animation APIs |
| 60 | Client-Side Storage |
|  | **Performance** |
| 61 | Performance Basics |
| 62 | Improving Page Rendering |
| 63 | Measuring Performance |
| 64 | CSS and Performance |
| 65 | JavaScript and Performance |
|  | **Security and Privacy** |
| 66 | Security and Privacy Basics |
| 67 | Data Protection Laws |
|  | **Testing** |
| 68 | General Testing Fundamentals |
| 69 | Functional and Compact Testing |
| 70 | Usability Testing |
| **71** | **JavaScript Frameworks** |
|  | **CSS Tooing** |
| 72 | CSS Frameworks |
| 73 | CSS Preprocessors |
| **74** | **Other Tooling Types** |

## **MDN Getting Stared Modules**

|  |  |
| --- | --- |
|  | **Getting Started Modules** |
|  | **Learn Web Development** |
| 01 | The Essential Skillset for New Front-End Developers |
| 02 | About Learn Web Development |
| 03 | Don’t Know Where to Get Stated? |
| 04 | Getting Our Code Examples |
| 05 | Contact Us |
| 06 | See Also |
|  | **Getting Started Modules** |
| 07 | Prerequisites |
| 08 | Modules |
| 09 | Environment Setup |
| 10 | Your First Website |
| 11 | Web Standards |
| 12 | Soft Skills |
|  | **Module-01: Environment Setup** |
|  | **Installing Basic Software** |
| 13 | Overview: Environment Setup |
| 14 | Installing a Code Editor |
| 15 | Installing Modern Web Browsers |
| 16 | Installing a Local Web Server |
| 17 | What Tools Do The Professionals Use? |
|  | **Browsing The Web** |
| 18 | How the Web Works: The Basics |
| 19 | The Difference Between Web Page, Website, Web Server, and Search Engine |
| 20 | Searching for Information |
|  | **Code Editors** |
| 21 | What Code Editors are Available? |
| 22 | Basic Code Editor Functionality |
| 23 | Enhancing Your Code Editor with Extensions |
|  | **Dealing with Files** |
| 24 | Where Should Your Website Live on Your Computer? |
| 25 | An Aside on Casing and Spacing |
| 26 | What Structure Should a Website Have? |
| 27 | File Paths |
|  | **Command Line Crash Course** |
| 28 | Welcome to The Terminal |
| 29 | Basic Built-in Terminal Commands |
| 30 | Terminal – Considered Harmful? |
| 31 | Connecting Commands Together with Pipes |
| 32 | A Slightly More Complex Example |
| 33 | Adding Powerups |
| 34 | Other Tools to Play with |
|  | **Module-02: Your First Website** |
|  | **What will Your Website Look Like?** |
| 35 | First Things First: Planning |
| 36 | Sketching Out Your Design |
| 37 | Choosing a Theme Color |
| 38 | Choosing an Image |
| 39 | Choosing a Font |
|  | **HTML: Creating the Content** |
| 40 | So What is HTML? |
| 41 | Creating Your First HTML Document |
| 42 | Images |
| 43 | Marking up Text |
| 44 | Links |
| 45 | Conclusion |
|  | **CSS: Styling the Content** |
| 46 | What is CSS? |
| 47 | Fonts and Text |
| 48 | CSS: All About Boxes |
| 49 | Conclusion |
|  | **JavaScript: Adding Interactivity** |
| 50 | What is JavaScript? |
| 51 | A “Hello World!” Example |
| 52 | Language Basics Crash Course |
| 53 | Supercharging Our Example Website |
| 54 | Conclusion |
|  | **Publishing Your Website** |
| 55 | What are the Options? |
| 56 | Publishing via GitHub |
| 57 | Further Reading |
|  | **Module-03: Web Standards** |
|  | **How the Web Works** |
| 58 | Clients and Servers |
| 59 | The Other Parts of the Toolbox |
| 60 | So What Happens, Exactly? |
| 61 | Other in Which Component Files are Parsed |
| 62 | DNS Explained |
| 63 | Packets Explained |
| 64 | See Also |
|  | **The Web Standards Model** |
| 65 | Brief History of the Web |
| 66 | Web Standards |
| 67 | Standards Bodies |
| 68 | The Web Standards Lifecycle |
| 69 | Overview of Modern Web Technologies |
| 70 | Web Best Practices |
| 71 | See Also |
|  | **How Browsers Load Websites** |
| 72 | The Different Types of File Returned in an HTTP Response |
| 73 | Static Versus Dynamic Files |
| 74 | How Files are Assembled to Render a Web Document in the Browser |
| 75 | The Browser: Hostile Versus Awesome Programming Environment |
|  | **Module-04: Soft Skills** |
|  | **Research and Learning** |
| 76 | Effective Research |
| 77 | Constant Learning Mindset |
| 78 | See Also |
|  | **Collaboration and Teamwork** |
| 79 | Getting on with Others |
| 80 | Participate, Help People, Ask Questions |
| 81 | Learn to Embrace Failure |
|  | **Workflows and Processes** |
| 82 | Typical Technology Combinations and Application Architectures |
| 83 | Typical Processes for a Technical Project |
| 84 | Typical Job Roles |
| 85 | Common Work Management |
| 86 | See Also |
|  | **Succeeding in Job Interviews** |
| 87 | Effective Strategies for Job Searching |
| 88 | Developing Qualities for That Employers Look for |
| 89 | Finding Work Experience |
| 90 | Selling Yourself to Would-be Employers |
| 91 | Preparing Yourself for Job Interviews |
| 92 | See Also |

# **Front-End Development**

## **MDN Core Modules: HTML**

|  |  |
| --- | --- |
|  | **Core Modules** |
|  | **Core Modules** |
| 01 | Prerequisites |
| 02 | Modules |
| 03 | Structuring Content with HTML |
| 04 | CSS Styling Basics |
| 05 | CSS Text Styling |
| 06 | CSS Layout |
| 07 | Dynamic Scripting with JavaScript |
| 08 | JavaScript Frameworks and Libraries |
| 09 | Accessibility |
| 10 | Design for Developers |
| 11 | Version Control |
|  | **Core Modules** |
|  | **Module-01: Structuring Content with HTML** |
|  | **Basic HTML Syntax** |
| 01 | What is HTML? |
| 02 | Anatomy of an HTML Element |
| 03 | Attributes |
| 04 | Anatomy of an HTML Document |
| 05 | Character References: Including Special Characters in HTLM |
| 06 | HTML Comments |
| 07 | Summary |
| 08 | See Also |
|  | **What’s in the Head? Webpage Metadata** |
| 09 | What is the HTML Head? |
| 10 | Adding a Title |
| 11 | Metadata: The <meta> Element |
| 12 | Adding Custom Icons to Your Site |
| 13 | Applying CSS and JavaScript to HTML |
| 14 | Setting the Primary Language of the Document |
| 15 | Summary |
|  | **Headings and Paragraphs in HTML** |
| 16 | Headings and Paragraphs |
| 17 | Implementing Structural Hierarchy |
| 18 | Why do We Need Structure? |
| 19 | Active Learning: Giving Our Content Structure |
| 20 | Why Do We Need Semantics? |
| 21 | Summary |
|  | **Emphasis and Importance** |
| 22 | What are Emphasis and Importance? |
| 23 | Active Learning: Let’s Be Important |
| 24 | Italic, Bold, Underline … |
| 25 | Summary |
|  | **Lists** |
| 26 | Unordered Lists |
| 27 | Ordered |
| 28 | Active Learning: Marking up Our Recipe Page |
| 29 | Nesting Lists |
| 30 | Description Lists |
| 31 | Test Your Skills |
| 32 | Summary |
|  | **Structuring Documents** |
| 33 | Basic Sections of a Document |
| 34 | HTML for Structuring Content |
| 35 | HTML Layout Elements in More Detail |
| 36 | Planning a Simple Website |
| 37 | Summary |
|  | **Advanced Text Features** |
| 38 | Quotations |
| 39 | Abbreviations |
| 40 | Marking up Contact Details |
| 41 | Superscript and Subscript |
| 42 | Representing Computer Code |
| 43 | Marking up Times and Dates |
| 44 | Test Your Skills |
| 45 | Summary |
|  | **Creating Links** |
| 46 | What is a Hyperlink? |
| 47 | Anatomy of a Link |
| 48 | A Quick Primer on URLs and Paths |
| 49 | Link Best Practices |
| 50 | Active Learning: Creating a Navigation Menu |
| 51 | Email Links |
| 52 | Test Your Skills |
| 53 | Summary |
|  | **Challenge: Marking up a Letter** |
| 54 | Starting Point |
| 55 | Project Brief |
| 56 | Hints and Tips |
| 57 | Example |
|  | **Challenge: Structuring a Page of Content** |
| 58 | Starting Point |
| 59 | Project Brief |
| 60 | Hints and Tips |
|  | **HTML Images** |
| 61 | How Do We Put an Image on a Webpage? |
| 62 | Media Assets and Licensing |
| 63 | Annotating Images with Figures and Figure Captions |
| 64 | CSS Background Images |
| 65 | Test Your Skills |
| 66 | Summary |
|  | **HTML Video and Audio** |
| 67 | Video and Audio on the Web |
| 68 | The <Video> Element |
| 69 | Using Multiple Source Formats to Improve Compatibility |
| 70 | Other <video> Features |
| 71 | The <audio> Element |
| 72 | Displaying Video Text Tracks |
| 73 | Active Learning: Embedding Your Own Audio and Video |
| 74 | Test Your Skills |
| 75 | Summary |
|  | **Challenge: Mozilla Splash Page** |
| 76 | Starting Point |
| 77 | Project Brief |
| 78 | Hints and Tips |
| 79 | Example |
|  | **HTML Table Basics** |
| 80 | What is a Table? |
| 81 | Active Learning: Creating Your First Table |
| 82 | Adding Headers with <th> Elements |
| 83 | Allowing Cells to Span Multiple Rows and Columns |
| 84 | Summary |
|  | **HTML Table Accessibility** |
| 85 | Recap: Tables for Visually Impaired Users |
| 86 | Adding a Caption to Your Table with <caption> |
| 87 | Adding Structure with <thead>, <tbody> and <tfoot> |
| 88 | The (Scope) Attribute |
| 89 | The (Id) and (headers) Attributes |
| 90 | Active Learning: Playing with (scope and headers) |
| 91 | Summary |
|  | **Challenge: Structuring a Planet Data Table** |
| 92 | Starting Point |
| 93 | Project Brief |
| 94 | Hints and Tips |
| 95 | Example |
|  | **Forms and Buttons in HTML** |
| 96 | Buttons |
| 97 | Common Input Types |
| 98 | Accessible Forms |
| 99 | Other Control Types |
| 100 | Client-Side Form Validation |
| 101 | The <form> Element |
| 102 | Test Your Skills |
| 103 | Summary |
|  | **Debugging HTML** |
| 104 | Debugging Isn’t Scary |
| 105 | HTML and Debugging |
| 106 | Permissive Code |
| 107 | HTML Validation |
| 108 | Using a DOM Inspector |
| 109 | Summary |

## **MDN Core Modules: CSS**

|  |  |
| --- | --- |
|  | **Core Modules** |
|  | **Module-02: CSS Styling Basics** |
|  | **What is CSS?** |
| 01 | Browser Default Styles |
| 02 | What is CSS for? |
| 03 | CSS Syntax Basics |
| 04 | How is CSS Applied to HTML? |
| 05 | Summary |
|  | **Getting Started with CSS** |
| 06 | Starting with Some HTML |
| 07 | Adding CSS to Our Document |
| 08 | Using Common Selectors |
| 09 | Other CSS Syntax Features |
| 10 | Summary |
|  | **Challenge: Styling a Biography Page** |
| 11 | Project Brief |
| 12 | Hints and Tips |
| 13 | Example |
|  | **Basic CSS Selectors** |
| 14 | What is a Selector? |
| 15 | Type Selectors |
| 16 | Class Selectors |
| 17 | ID Selectors |
| 18 | Selector Lists |
| 19 | The Universal Selector |
| 20 | Summary |
|  | **Attribute Selectors** |
| 21 | Presence and Value Selectors |
| 22 | Substring Matching Selectors |
| 23 | Summary |
|  | **Pseudo-Classes and Pseudo-Element** |
| 24 | What is a Pseudo-Class? |
| 25 | What is a Pseudo-Element? |
| 26 | Combining Pseudo-Classes and Pseudo-Elements |
| 27 | Generating Content with ::before and ::after |
| 28 | Summary |
| 29 | See Also |
|  | **Combinators** |
| 30 | Descendant Combinator |
| 31 | Child Combinator |
| 32 | Next-Sibling Combinator |
| 33 | Subsequent-Sibling Combinator |
| 34 | Creating Complex Selectors with Nesting |
| 35 | Combining Combinators with Selectors |
| 36 | Test Your Skills |
| 37 | Summary |
|  | **The Box Model** |
| 38 | Block and Inline Boxes |
| 39 | Inner and Outer Display Types |
| 40 | Examples of Different Display Types |
| 41 | What is the CSS Box Model? |
| 42 | Playing with Box Models |
| 43 | Margins, Padding, and Borders |
| 44 | The Box Model and Inline Boxes |
| 45 | Using Display: Inline-Block |
| 46 | Test Your Skills |
| 47 | Summary |
|  | **Handling Conflicts** |
| 48 | Conflicting Rules |
| 49 | Understanding Inheritance |
| 50 | Understanding the Cascade |
| 51 | The Effect of CSS Location |
| 52 | Test Your Skills |
| 53 | Summary |
|  | **CSS Values and Units** |
| 54 | What is a CSS Value? |
| 55 | Numbers, Lengths, and Percentages |
| 56 | Color |
| 57 | Images |
| 58 | Position |
| 59 | Strings and Identifiers |
| 60 | Functions |
| 61 | Test Your Skills |
| 62 | Summary |
|  | **Sizing Items in CSS** |
| 63 | The Natural or Intrinsic Size of Things |
| 64 | Setting a Specific Size |
| 65 | min- and max- sizes |
| 66 | Viewport units |
| 67 | Test Your Skills |
| 68 | Summary |
|  | **Backgrounds and Borders** |
| 69 | Styling Backgrounds in CSS |
| 70 | Background Colors |
| 71 | Background Images |
| 72 | Gradient Backgrounds |
| 73 | Multiple Background Images |
| 74 | Background Attachment |
| 75 | Using the Background Shorthand Property |
| 76 | Accessibility Considerations with Backgrounds |
| 77 | Borders |
| 78 | Rounded Corners |
| 79 | Test Your Skills |
| 80 | Summary |
|  | **Overflowing Content** |
| 81 | What is Overflow? |
| 82 | CSS Tries to Avoid “Data Loss” |
| 83 | The Overflow Property |
| 84 | Unwanted Overflow in Web Design |
| 85 | Test Your Skills |
| 86 | Summary |
|  | **Images, Media, and Form Elements** |
| 87 | Replaced Elements |
| 88 | Sizing Images |
| 89 | Replaced Elements in Layout |
| 90 | Form Elements |
| 91 | Styling Text Input Elements |
| 92 | Normalizing Form Behavior |
| 93 | Test Your Skills |
| 94 | Summary |
|  | **Styling Tables** |
| 95 | A Typical HTML Table |
| 96 | Getting Started with Styling Our Table |
| 97 | Spacing and Layout |
| 98 | Some Simple Typography |
| 99 | Graphics and Colors |
| 100 | Styling the Caption |
| 101 | Table Styling Quick Tips |
| 102 | Test Your Skills |
| 103 | Summary |
|  | **Debugging CSS** |
| 104 | How to Access Browser DevTools |
| 105 | The DOM Versus View Source |
| 106 | Inspecting the Applied CSS |
| 107 | Editing Values |
| 108 | Adding a New Property |
| 109 | Understanding the Box Model |
| 110 | Solving Specificity Issues |
| 111 | Debugging Problems in CSS |
| 112 | Summary |
| 113 | See Also |
|  | **Challenge: Fundamental CSS Comprehension** |
| 114 | Starting Point |
| 115 | Project Brief |
| 116 | Hints and Tips |
| 117 | Example |
|  | **Challenge: Creating Fancy Letter-Headed Paper** |
| 118 | Starting Point |
| 119 | Project Brief |
| 120 | Hints and Tips |
| 121 | Example |
|  | **Challenge: A Cool-Looking Box** |
| 122 | Starting Point |
| 123 | Project Brief |
| 124 | Hints and Tips |
| 125 | Example |
|  | **Module-03: CSS Text Styling** |
|  | **Fundamental Text and Font Styling** |
| 126 | What is Involved in Styling Text in CSS? |
| 127 | Fonts |
| 128 | Text Layout |
| 129 | Font Shorthand |
| 130 | Active Learning: Playing with Styling Text |
| 131 | Summary |
| 132 | See Also |
|  | **Styling Lists** |
| 133 | A Simple List Example |
| 134 | Handling List Spacing |
| 135 | List-Specific Styles |
| 136 | Controlling List Counting |
| 137 | Active Learning: Styling a Nested List |
| 138 | Summary |
|  | **Styling Links** |
| 139 | Link States |
| 140 | Default Styles |
| 141 | Styling Links |
| 142 | Active Learning: Style Your Own links |
| 143 | Including Icons on Links |
| 144 | Styling Links As Buttons |
| 145 | Summary |
|  | **Web Fonts** |
| 146 | Font Families Recap |
| 147 | Web Fonts |
| 148 | Active Learning: A Web Font Example |
| 149 | Using an Online Font Service |
| 150 | @font-face in More Detail |
| 151 | Summary |
| 152 | See Also |
|  | **Challenge: Typesetting a Community School Homepage** |
| 153 | Starting Point |
| 154 | Project Brief |
| 155 | Hints and Tips |
| 156 | Example |
|  | **Module-04: CSS Layout** |
|  | **Introduction to CSS Layout** |
| 157 | Normal Layout Flow |
| 158 | How are Elements Laid Out by Default? |
| 159 | Overriding Normal Flow |
| 160 | Summary |
|  | **Floats** |
| 161 | The Background of Floats |
| 162 | A (float) Example |
| 163 | Clearing (floats) |
| 164 | Clearing Boxes Wrapped Around a (float) |
| 165 | Test Your Skills |
| 166 | Summary |
|  | **Positioning** |
| 167 | Introducing Positioning |
| 168 | Static Positioning |
| 169 | Relative Positioning |
| 170 | Absolute Positioning |
| 171 | Fixed Positioning |
| 172 | Sticky Positioning |
| 173 | Test Your Skills |
| 174 | Summary |
|  | **Flexbox** |
| 175 | Why Flexbox? |
| 176 | Introducing a Simple Example |
| 177 | Specifying What Elements to Lay Out As Flexible Boxes |
| 178 | The Flex Model |
| 179 | Columns or Rows? |
| 180 | Wrapping |
| 181 | (Flex-flow) Shorthand |
| 182 | Flexible Sizing of (flex) Items |
| 183 | Flex: Shorthand Versus Longhand |
| 184 | Horizontal and Vertical Alignment |
| 185 | Ordering (flex) Items |
| 186 | Nested (flex) Boxes |
| 187 | Test Your Skills |
| 188 | Summary |
|  | **CSS Grid Layout** |
| 189 | What is Grid Layout? |
| 190 | Creating Your Grid in CSS |
| 191 | Line-Based Placement |
| 192 | Positioning with grid-template-areas |
| 193 | Nesting grids and subgrid |
| 194 | Grid Systems |
| 195 | Test Your Skills |
| 196 | Summary |
| 197 | See Also |
|  | **Responsive Design** |
| 198 | Precursor to Responsive Design: Mobile Web Design |
| 199 | Introducing Responsive Web Design |
| 200 | Media Queries |
| 201 | Responsive Layout Technology |
| 202 | Responsive Images/Media |
| 203 | Responsive Typography |
| 204 | The (viewport) Meta Tag |
| 205 | Summary |
| 206 | See Also |
|  | **Media Query Fundamentals** |
| 207 | Media Query Basics |
| 208 | More Complex Media Queries |
| 209 | How to Choose Breakpoints |
| 210 | Active Learning: Mobile First Responsive Design |
| 211 | The (viewport) Meta Tab |
| 212 | Do You Really Need a Media Query? |
| 213 | Test Your Skills |
| 214 | Summary |
| 215 | Challenge: Fundamental Layout Comprehension |
| 216 | Starting Point |
| 217 | Project Brief | Hints and Tips | Examples |

## **MDN Core Modules: JavaScript**

|  |  |
| --- | --- |
|  | **Core Modules** |
|  | **Module-05: Dynamic Scripting with JavaScript** |
|  | **What is JavaScript** |
| 01 | A High-Level Definition |
| 02 | So What Can It Really Do? |
| 03 | What is JavaScript Doing On Your Page? |
| 04 | How Do You Add JavaScript to Your Page? |
| 05 | Comments |
| 06 | Summary |
|  | **A First Splash Into JavaScript** |
| 07 | Thinking Like a Programmer |
| 08 | Example- Guess The Number Game |
| 09 | Summary |
|  | **What Went Wrong? Troubleshooting JavaScript** |
| 10 | Types of Error |
| 11 | An Erroneous Example |
| 12 | Fixing Syntax Errors |
| 13 | A logic Error |
| 14 | Other Common Errors |
| 15 | See Also |
|  | **Storing The Information You Need – Variables** |
| 16 | Tools You Need |
| 17 | What is a Variable? |
| 18 | Declaring a Variable |
| 19 | Initializing a Variable |
| 20 | A Note About (var) |
| 21 | Updating a Variable |
| 22 | Variable Types |
| 23 | Dynamic Typing |
| 24 | Constants in JavaScript |
| 25 | When to Use (const) and When to Use (let) |
| 26 | Test Your Skills |
| 27 | Summary |
|  | **Basic Math in JavaScript – Numbers and Operators** |
| 28 | Everybody Loves Math |
| 29 | Arithmetic Operators |
| 30 | Increment and Decrement Operators |
| 31 | Assignment Operators |
| 32 | Active Learning: Sizing a Canvas Box |
| 33 | Comparison Operators |
| 34 | Test Your Skills |
| 35 | Summary |
|  | **Handling Text – Strings in JavaScript** |
| 36 | The Poser of Words |
| 37 | Declaring Strings |
| 38 | Embedding JavaScript |
| 39 | Multiline Strings |
| 40 | Including Quotes in Strings |
| 41 | Numbers vs. Strings |
| 42 | Summary |
|  | **Useful String Methods** |
| 43 | Strings as Objects |
| 44 | Finding the Length of a String |
| 45 | Retrieving a Specific String Character |
| 46 | Testing if a String Contains a Substring |
| 47 | Finding the Position of a Substring in a String |
| 48 | Extracting a Substring from a String |
| 49 | Changing Case |
| 50 | Updating Parts of a String |
| 51 | Active Learning Examples |
| 52 | Test Your Skills |
| 53 | Conclusion |
|  | **Arrays** |
| 54 | What is an Array? |
| 55 | Creating Arrays |
| 56 | Finding the Length of an Array |
| 57 | Accessing and Modifying Array Items |
| 58 | Finding the Index of Items in an Array |
| 59 | Adding Items |
| 60 | Removing Items |
| 61 | Accessing Every Item |
| 62 | Converting Between Strings and Arrays |
| 63 | Active Learning: Printing Those Products |
| 64 | Active Learning: Top 5 Searches |
| 65 | Test Your Skills |
| 66 | Conclusion |
|  | **Challenge: Silly Story Generator** |
| 67 | Starting Point |
| 68 | Project Brief |
| 69 | Steps to Complete |
| 70 | Hints and Tips |
|  | **Making Decisions in Your Code – Conditionals** |
| 71 | You Can Have It On One Condition |
| 72 | If…else Statements |
| 73 | Switch Statements |
| 74 | Ternary Operator |
| 75 | Active Learning: A Simple Calendar |
| 76 | Active Learning: More Color Choices |
| 77 | Test Your Skills |
| 78 | Summary |
| 79 | See Also |
|  | **Looping Code** |
| 80 | Why are Loops Useful? |
| 81 | Looping Through a Collection |
| 82 | The Standard (for) Loop |
| 83 | Exiting Loops with (break) |
| 84 | Skipping Iterations with Continue |
| 85 | (while) and (do…while) |
| 86 | Active Learning: Launch Countdown |
| 87 | Active Learning: Filling in a Guest List |
| 88 | Which Loop Type Should You Use? |
| 89 | Test Your Skills |
| 90 | Summary |
| 91 | See Also |
|  | **Functions – Reusable Blocks of Code** |
| 92 | Where Do I Find Functions? |
| 93 | Built-in Browser Functions |
| 94 | Functions Versus Methods |
| 95 | Invoking Functions |
| 96 | Function Parameters |
| 97 | Anonymous Functions and Arrow Functions |
| 98 | Function Scope and Conflicts |
| 99 | Test Your Skills |
| 100 | Summary |
| 101 | See Also |
|  | **Build Your Own Function** |
| 102 | Active Learning: Let’s Build a Function |
| 103 | The Basic Function |
| 104 | Calling the Function |
| 105 | Improving the Function with Parameters |
| 106 | Test Your Skills |
| 107 | Summary |
|  | **Function Return Values** |
| 108 | What are Return Values? |
| 109 | Using Return Values in Your Own Functions |
| 110 | Active Learning: A Return Value Function |
| 111 | Now It’s Your Turn |
| 112 | Test Your Skills |
| 113 | Conclusion |
| 114 | See Also |
|  | **Introduction to Events** |
| 115 | What is an Event? |
| 116 | Using addEventListener() |
| 117 | Other Event Listener Mechanisms |
| 118 | Event Objects |
| 119 | Preventing Default Behavior |
| 120 | It’s Not Just Web Pages |
| 121 | Summary |
|  | **Event Bubbling** |
| 122 | Introducing Event Bubbling |
| 123 | Event Capture |
| 124 | Event Delegation |
| 125 | (target) and (currrentTarget) |
| 126 | Test Your Skills |
| 127 | Summary |
| 128 | See Also |
|  | **Challenge: Image Gallery** |
| 129 | Starting Point |
| 130 | Project Brief |
| 131 | Steps to Complete |
| 132 | Declare an Array of Image (filenames) |
| 133 | Hints and Tips |
|  | **JavaScript Object Basics** |
| 134 | Object Basics |
| 135 | Dot Notation |
| 136 | Bracket Notation |
| 137 | Setting Object Members |
| 138 | What is “this”? |
| 139 | Introducing Constructors |
| 140 | You’ve Been Using Objects All Along |
| 141 | Test Your Skills |
| 142 | Summary |
|  | **DOM Scripting Introduction** |
| 143 | The Important Parts of a Web Browser |
| 144 | The Document Object Model |
| 145 | Active Learning: Basic DOM Manipulation |
| 146 | Active Learning: A Dynamic Shopping List |
| 147 | Summary |
| 148 | See Also |
|  | **Making Network Requests with JavaScript** |
| 149 | What is the Problem Here? |
| 150 | The Fetch API |
| 151 | The XMLHttpRequest API |
| 152 | Summary |
| 153 | See Also |
|  | **Working with JSON** |
| 154 | No, Really, What is JSON? |
| 155 | Active Learning: working Through a JSON Example |
| 156 | Converting Between Object and Text |
| 157 | Test Your Skills |
| 158 | Summary |
| 159 | See Also |
|  | **Debugging JavaScript and Handling Errors** |
| 160 | Recap on Types of JavaScript Error |
| 161 | Linting Your Code |
| 162 | Common JavaScript Problems |
| 163 | The Browser JavaScript Console |
| 164 | The Console API |
| 165 | Using the JavaScript Debugger |
| 166 | Handling JavaScript Error in Your Code |
| 167 | Finding Help |
| 168 | Summary |
|  | **Module-06: JavaScript Frameworks and Libraries** |
| 169 | **Introduction to Client-Side Frameworks** |
| 170 | The Emergence of Libraries and Frameworks |
| 171 | What Frameworks are Out There? |
| 172 | Why Do Frameworks Exist? |
| 173 | The Verbosity of DOM Changes |
| 174 | Another Way to Build UIs |
| 175 | Other Things Frameworks Give Us |
| 176 | Things to Consider When Using Frameworks |
| 177 | Accessibility On a Framework-Driven Web |
| 178 | How to Choose a Framework |
| 179 | Alternatives to Client-Side Frameworks |
| 180 | Summary |
|  | **Framework Main Features** |
| 181 | Domain-Specific Languages |
| 182 | Writing Components |
| 183 | Styling Components |
| 184 | Handling Dependencies |
| 185 | Rendering Elements |
| 186 | Routing |
| 187 | Testing |
| 188 | Summary |
|  | **Getting Started with React** |
| 189 | Hello React |
| 190 | Use Cases |
| 191 | How Does React Use JavaScript? |
| 192 | Setting Up Your First React App |
| 193 | Exploring Our First React Component - <App /> |
| 194 | Moving on to (main) |
| 195 | Starting (fresh) |
| 196 | Practice with (JSX) |
| 197 | Summary |
| 198 | See Also |
|  | **Beginning Our React To-do List** |
| 199 | Our App’s User Stories |
| 200 | Pre-Project Housekeeping |
| 201 | Project Starter Code |
| 202 | Summary |
|  | **Componentizing Our React App** |
| 203 | Defining Our First Component |
| 204 | Make a <Todo /> |
| 205 | Make a Unique <Todo /> |
| 206 | So far, So Good? |
| 207 | Tasks as Data |
| 208 | Rendering with Iteration |
| 209 | Unique Keys |
| 210 | Componentizing the Rest of the App |
| 211 | Importing All Our Components |
| 212 | Summary |
|  | **React Interactivity: Events and State** |
| 213 | Handling Events |
| 214 | Callback Props |
| 215 | Persisting and Changing Data with State |
| 216 | Putting It All Together: Adding a Task |
| 217 | Detour: Counting Tasks |
| 218 | Completing a Task |
| 219 | Deleting a Task |
| 220 | Deleting Tasks From State and UI |
| 221 | Summary |
|  | **React Interactivity: Editing, Filtering, Conditional Rendering** |
| 222 | Editing the Name of a Task |
| 223 | A UI for Editing |
| 224 | Conditional Rendering |
| 225 | Toggling the <Todo /> templates |
| 226 | Editing from the UI |
| 227 | Back to the Filter Buttons |
| 228 | Summary |
|  | **Accessibility in React** |
| 229 | Including Keyboard Users |
| 230 | Exploring the Keyboard Usability Problem |
| 231 | Focusing Between Templates |
| 232 | More Robust Focus Management |
| 233 | Focusing When the User Deletes a Task |
| 234 | Finished |
|  | **React Resources** |
| 235 | Component-Level Styles |
| 236 | React DevTools |
| 237 | The (useReducer()) Hook |
| 238 | The Context API |
| 239 | Class components |
| 240 | Testing |
| 241 | Routing |

## **MDN Core Modules: Design**

|  |  |
| --- | --- |
|  | **Core Modules** |
|  | **Module-07: Design for Developers** |
| 01 | Prerequisites |
| 02 | Basic Design Theory |
| 03 | User-Centered Design |
| 04 | Design Briefs |
| 05 | See Also |

# **Full-Stack Development**

## **MDN Extension Modules: Advance JavaScript**

|  |  |
| --- | --- |
|  | **Extension Modules** |
| 01 | Prerequisites |
| 02 | Modules |
| 03 | * Advanced JavaScript Objects |
| 04 | * Asynchronous JavaScript |
| 05 | * Web Forms |
| 06 | * Server-Side Website Programming |
| 07 | * Client-Side Web APIs |
| 08 | * Understanding Client-Side Web Development Tools |
| 09 | * Web Performance |
| 10 | * Testing |
| 11 | * Transform and Animation CSS |
| 12 | * Security and Privacy |
|  | **Extension Modules** |
|  | **Module-01: Advanced JavaScript Objects** |
|  | **Object Prototypes** |
| 01 | The Prototype Chain |
| 02 | Shadowing Properties |
| 03 | Setting a Prototype |
| 04 | Prototypes and Inheritance |
| 05 | Summary |
|  | **Object-Oriented Programming** |
| 06 | Classes and Instances |
| 07 | Inheritance |
| 08 | Encapsulation |
| 09 | OOP and JavaScript |
| 10 | Summary |
|  | **Classes in JavaScript** |
| 11 | Classes and Constructors |
| 12 | Inheritance |
| 13 | Encapsulation |
| 14 | Test Your Skills |
| 15 | Summary |
|  | **Object Building Practice** |
| 16 | Let’s Bounce Some Balls |
| 17 | Getting Stared |
| 18 | Modeling a Ball in Our Program |
| 19 | Animation the Ball |
| 20 | Adding Collision Detection |
| 21 | Summary |
| 22 | See Also |
|  | **Challenge: Adding Features to Our Bouncing Balls Demo** |
| 23 | Starting Point |
| 24 | Hints and Tips |
| 25 | Project Brief |
| 26 | Steps to Complete |
|  | **Module-02: Asynchronous JavaScript** |
|  | **Introducing Asynchronous JavaScript** |
| 27 | Synchronous Programming |
| 28 | A Long-Running Synchronous Function |
| 29 | The Trouble with Log-Running Synchronous Functions |
| 30 | Event Handlers |
| 31 | Callbacks |
|  | **Video and Audio APIs (Above)** |
|  | **How to Use Promises** |
| 32 | Using the (fetch()) API |
| 33 | Chaining Promises |
| 34 | Catching Errors |
| 35 | Promise Terminology |
| 36 | Combining Multiple Promises |
| 37 | (async) and (await) |
| 38 | Summary |
| 39 | See Also |
|  | **How to Implement a Promise-based API** |
| 40 | Implementing an (alarm()) API |
| 41 | Using the (alarm()) API |
| 42 | Using (async) and (await) with the (alarm()) API |
| 43 | See Also |
|  | **Introducing Workers** |
| 44 | Using Web Workers |
| 45 | Other Types of Workers |
| 46 | Summary |
| 47 | See Also |
|  | **Challenge: Sequencing Animations** |
| 48 | Starting Point |
| 49 | Project Brief |
| 50 | Steps to Complete |
|  | **Module-03: Web Forms** |
|  | **Your First Form** |
| 51 | What are Web Forms? |
| 52 | Designing Your Form |
| 53 | Active Learning: Implementing Our Form HTML |
| 54 | Basic Form Styling |
| 55 | Sending Form Data to Your Web Server |
| 56 | Summary |
|  | **How to Structure a Web Form** |
| 57 | The <form> Element |
| 58 | The <fieldset> and <legend> Elements |
| 59 | The <label> Element |
| 60 | Common HTML Structures Used with Forms |
| 61 | Test Your Skills |
| 62 | Summary |
| 63 | See Also |
|  | **Basic Native Form Controls** |
| 64 | Text Input Fields |
| 65 | Checkable Items: Checkboxes and Radio Buttons |
| 66 | Actual Buttons |
| 67 | File Picker |
| 68 | Common Attributes |
| 69 | Test Your Skills |
| 70 | Summary |
|  | **The HTML5 Input Types** |
| 71 | Email Address Field |
| 72 | Search Field |
| 73 | Phone Number Field |
| 74 | URL Field |
| 75 | Numeric Field |
| 76 | Slider Controls |
| 77 | Date and Time Pickers |
| 78 | Color Picker Control |
| 79 | Test Your Skills |
| 80 | Summary |
|  | **Other Form Controls** |
| 81 | Multi-Line Text Fields |
| 82 | Drop-Down Controls |
| 83 | Other Form Features |
| 84 | Test Your Skills |
| 85 | Summary |
|  | **Styling Web Forms** |
| 86 | Challenges in Styling Form Widgets |
| 87 | Styling Simple Form Widgets |
| 88 | A Specific Styling Example |
| 89 | Test Your Skills |
| 90 | Summary |
|  | **Advanced Form Styling** |
| 91 | Appearance: Controlling OS-Level Styling |
| 92 | What Can Be Done About The “ugly” Elements? |
| 93 | Summary |
|  | **UI Pseudo-Classes** |
| 94 | What Pseudo-Classes Do We Have Available? |
| 95 | Styling Inputs Based On Whether They are Required or Not |
| 96 | Using Generated Content with Pseudo-Classes |
| 97 | Styling Controls Based On Whether Their Data is Valid |
| 98 | Styling Enabled and Disabled Inputs, and Read-Only and Read-Write |
| 99 | Radio and Checkbox State – Checked, Default, Indeterminate |
| 100 | More Pseudo-Classes |
| 101 | Test Your Skills |
| 102 | Summary |
|  | **Client-Side Form Validation** |
| 103 | What is Form Validation? |
| 104 | Different Types of Client-Side Validation |
| 105 | Using Build-in Form Validation |
| 106 | Built-in Form Validation Examples |
| 107 | Validating Form Using JavaScript |
| 108 | Test Your Skills |
| 109 | Summary |
|  | **Sending Form Data** |
| 110 | Client/Server Architecture |
| 111 | On the Client Side: Defining How to Send the Data |
| 112 | On the Server Side: Retrieving the Data |
| 113 | A Special Case: Sending Files |
| 114 | Security Issues |
| 115 | Summary |
| 116 | See Also |
|  | **Module-04: Server-Side Websites** |
|  | **Server-Side First Steps** |
|  | **Introduction to the Server Side** |
| 117 | What is Server-Side Website Programming? |
| 118 | Are Server-Side and Client-Side Programming ths Same? |
| 119 | What Can You Do On the Server-Side? |
| 120 | Summary |
|  | **Client-Server Overview** |
| 121 | Web Servers and HTTP (A Primer) |
| 122 | Static Sites |
| 123 | Dynamic Sites |
| 124 | Web Frameworks Simplify Server-Side Web Programming |
| 125 | Summary |
|  | **Server-Side Web Frameworks** |
| 126 | Overview |
| 127 | What Can A Web Framework Do for You? |
| 128 | How to Select A Web Framework |
| 129 | A Few Good Web Frameworks? |
| 130 | Summary |
|  | **Web Security** |
| 131 | What is Website Security? |
| 132 | Website Security Threats |
| 133 | A Few Key Messages |
| 134 | Summary |
|  | **Express Web Framework (Node.JS)** |
|  | **Express/Node Introduction** |
| 135 | Introducing Node |
| 136 | Web Frameworks |
| 137 | Introducing Express |
| 138 | Where Did Node and Express Come From? |
| 139 | How Popular are Node and Express? |
| 140 | Is Express Opinionated? |
| 141 | What Does Express Code Look Like? |
| 142 | Summary |
|  | **Setting Up a Node Development Environment** |
| 143 | Express Development Environment Overview |
| 144 | Installing Node |
| 145 | Using NPM |
| 146 | Installing the Express Application Generator |
| 147 | Summary |
| 148 | See Also |
|  | **Express Tutorial: The Local Library Website** |
| 149 | Overview |
| 150 | The LocalLibrary Website |
| 151 | I’m stuck, Where Can I Get the Source? |
| 152 | Summary |
|  | **Express Tutorial Part-2: Creating a Skeleton Website** |
| 153 | Overview |
| 154 | Using the Application Generator |
| 155 | Creating the Project |
| 156 | Running the Skeleton Website |
| 157 | Enable Server Restart on File Changes |
| 158 | The Generated Project |
| 159 | Challenge Yourself |
| 160 | Summary |
| 161 | See Also |
|  | **Express Tutorial Part-3: Using a Database (with Mongoose)** |
| 162 | Overview |
| 163 | Designing the LocalLibrar Models |
| 164 | Mongoose Primer |
| 165 | Setting Up the MongoDB Database |
| 166 | Install Mongoose |
| 167 | Connect to MongoDB |
| 168 | Defining the LocalLibrary Schema |
| 169 | Testing – Create Some Items |
| 170 | Summary |
| 171 | See Also |
|  | **Express Tutorial Part-4: Routes and Controllers** |
| 172 | Overview |
| 173 | Routes Primer |
| 174 | Routes Needed for the LocalLibrary |
| 175 | Create the Route-Handler Callback Functions |
| 176 | Create the Catalog Route Module |
| 177 | Summary |
| 178 | See Also |
| 179 | **Express Tutorial Part-5: Displaying Library Data** |
| 180 | Overview |
| 181 | Displaying Library Data Tutorial Sub-articles |
| 182 | Summary |
| 183 | See Also |
|  | **Express Tutorial Par-6: Working with Forms** |
| 184 | Overview |
| 185 | Express Forms Sub-articles |
| 186 | Challenge Yourself |
| 187 | Summary |
| 188 | See Also |
|  | **Express Tutorial Part-7: Deploying to Production** |
| 189 | Overview |
| 190 | What is a Production Environment? |
| 191 | Choosing a Hosting Provider |
| 192 | Getting Your Website Ready to Publish |
| 193 | Example: Hosting on Glitch |
| 194 | Example: Hosting on Railway |
| 195 | Summary |
| 196 | See Also |
| 197 | **Django Web Framework (Python)** |

## **MDN Extension Modules: Advance Client-Side and Security**

|  |  |
| --- | --- |
|  | **Extension Modules** |
|  | **Module-01: Client-Side Web APIs** |
|  | **Introduction to Web APIs** |
| 01 | What are APIs? |
| 02 | What Can APIs Do? |
| 03 | How Do APIs Work? |
| 04 | Summary |
|  | **Video and Audio APIs** |
| 05 | HTML Video and Audio |
| 06 | The HTMLMediaElement API |
| 07 | Summary |
| 08 | See Also |
|  | **Drawing Graphics** |
| 09 | Graphics on the Web |
| 10 | Active Learning: Getting Started with a <canvas> |
| 11 | 2D Canvas Basics |
| 12 | Loops and Animations |
| 13 | WebGL |
| 14 | Summary |
| 15 | See Also |
| 16 | Example |
|  | **Client-Side Storage** |
| 17 | Client-Side Storage? |
| 18 | Storing Simple Data – Web Storage |
| 19 | Storing Complex Data – IndexedDB |
| 20 | Offline Asset Storage |
| 21 | Summary |
| 22 | See Also |
|  | **Third-Party APIs** |
| 23 | What are Third Party APIs? |
| 24 | Extending the MapQuest Example |
| 25 | A RESTful API – NYTimes |
| 26 | YouTube Example |
| 27 | Summary |
|  | **Module-02: Understanding Client-Side Tools** |
|  | **Client-Side Tooling Overview** |
| 28 | Overview of Modern Tooling |
| 29 | The Modern Tooling Ecosystem |
| 30 | How to Choose and Get Help with a Particular Tool |
| 31 | Summary |
|  | **Package Management Basics** |
| 32 | A Dependency in Your Project |
| 33 | What Exactly is a Package Manager? |
| 34 | Package Registries |
| 35 | Using the Package Ecosystem |
| 36 | A Rough Guide to Package Manager Clients |
| 37 | Making Your Own Commands |
| 38 | Summary |
| 39 | See Also |
|  | **Introducing a complete Toolchain** |
| 40 | Introducing Our Case Study |
| 41 | Tools Used in Our ToolChain |
| 42 | Toolchains and Their Inherent Complexity |
| 43 | Checking Prerequisites |
| 44 | Creating a Development Environment |
| 45 | Build and Transformation Tools |
| 46 | Write the Source Code |
| 47 | Running the Transformation |
| 48 | Summary |
|  | **Deploying Our App** |
| 49 | Post Development |
| 50 | The Build Process |
| 51 | Committing Changes to GitHub |
| 52 | Using GitHub Actions for Deployment |
| 53 | Testing |
| 54 | Summary |
|  | **Module-03: Transform and Animation CSS** |
| 55 | Prerequisites |
| 56 | Learning Outcomes |
| 57 | Guide |
| 58 | * Using CSS Transforms |
| 59 | * Using CSS Transitions |
| 60 | * Using CSS Animations |
|  | **Module-04: Security and Privacy** |
| 61 | Prerequisites |
| 62 | Learning Outcomes |
| 63 | Resources |
| 64 | See Also |

## **MDN Extension Modules**

|  |  |
| --- | --- |
|  | **Extension Modules** |
|  | **Module-01: Version Control** |
| 01 | Overview |
| 02 | Prerequisites |
| 03 | Learning Outcomes |
| 04 | Guides |
| 05 | Hello, World (from GitHub) |
| 06 | Git Handbook (from GitHub) |
| 07 | Forking Projects (from GitHub) |
| 08 | About Pull Requests (from GitHub) |
| 09 | Mastering Issues (from GitHub) |
| 10 | See Also |
|  | **Module-02: Accessibility** |
|  | **What is Accessibility?** |
| 11 | So What is Accessibility? |
| 12 | What Kinds of Disability are We Looking At? |
| 13 | Implementing Accessibility into Your Project |
| 14 | Accessibility Guidelines and the Law |
| 15 | Accessibility APIs |
| 16 | Summary |
| 17 | See Also |
|  | **Accessibility Tooling and Assistive Technology** |
| 18 | Accessibility Tools |
| 19 | Screen Readers |
| 20 | User Testing |
| 21 | Other Tooling |
| 22 | Accessibility Testing checklist |
| 23 | Summary |
|  | **HTML: A Good Basics for Accessibility** |
| 24 | HTML and Accessibility |
| 25 | Good Semantics |
| 26 | Accessible Data Tables |
| 27 | Text Alternatives |
| 28 | More on Links |
| 29 | Test Your Skills |
| 30 | Summary |
|  | **CSS and JavaScript Accessibility Best Practices** |
| 31 | CSS and JavaScript are Accessible? |
| 32 | CSS |
| 33 | JavaScript |
| 34 | Test Your Skills |
| 35 | Summary |
|  | **WAI-ARIA Basics** |
| 36 | What is WAI-ARIA? |
| 37 | Practical WAI-ARIA Implementations |
| 38 | Test Your Skills |
| 39 | Summary |
|  | **Accessible Multimedia** |
| 40 | Multimedia and Accessibility |
| 41 | Simple Images |
| 42 | Accessible Audio and Video Controls |
| 43 | Audio Transcripts |
| 44 | Video Text Tracks |
| 45 | Summary |
|  | **Mobile Accessibility** |
| 46 | Accessibility on Mobile Devices |
| 47 | Summary of Screen Reader Testing on Android and iOS |
| 48 | Control Mechanisms |
| 49 | Responsive Design |
| 50 | User Input |
| 51 | Summary |
| 52 | See Also |
|  | **Challenge: Accessibility Troubleshooting** |
| 53 | Starting Point |
| 54 | Project Brief |
|  | **Module-03: Web Performance** |
|  | **The “why” of Web Performance** |
| 55 | Why Care About Performance? |
| 56 | Conclusion |
|  | **What is Web Performance?** |
| 57 | What is Web Performance? |
| 58 | How Content is Rendered |
| 59 | Conclusion |
|  | **Perceived Performance** |
| 60 | Overview |
| 61 | Performance Metrics |
| 62 | Improving Performance |
| 63 | Conclusion |
|  | **Measuring Performance** |
| 64 | Performance Tools |
| 65 | General Performance Reporting Tools |
| 66 | Network Monitor Tools |
| 67 | Performance Monitor Tools |
| 68 | Performance APIs |
| 69 | Conclusion |
|  | **Multimedia: Images** |
| 70 | Why Optimize Your Multimedia? |
| 71 | Optimizing Image Delivery |
| 72 | Conclusion |
|  | **Multimedia: Video** |
| 73 | Why Optimize Your Multimedia? |
| 74 | Optimizing Video Delivery |
| 75 | Conclusion |
|  | **JavaScript Performance Optimization** |
| 76 | To Optimize or Not to Optimize |
| 77 | Optimizing JavaScript Downloads |
| 78 | Handling Parsing and Execution |
| 79 | Loading Critical Assets As Soon As Possible |
| 80 | Deferring Execution of Non-Critical JavaScript |
| 81 | Breaking Down Long Tasks |
| 82 | Handling JavaScript Animations |
| 83 | Optimizing Event Performance |
| 84 | Tips for Writing More Efficient Code |
| 85 | See Also |
|  | **HTML Performance Optimization** |
| 86 | To Optimize or Not to Optimize |
| 87 | Key HTML Performance Issues |
| 88 | Responsive Handling of Replaced Elements |
| 89 | Handling Embedded Content |
| 90 | Handling Resource Loading Order |
| 91 | See Also |
|  | **CSS Performance Optimization** |
| 92 | To Optimize or Not to Optimize |
| 93 | Optimizing Rendering |
| 94 | Handling Animations |
| 95 | Optimizing Element Changes with (will-change) |
| 96 | Optimizing for (render blocking) |
| 97 | Improving (font) performance |
| 98 | Optimizing Styling Recalculation with CSS Containment |
| 99 | See Also |
|  | **The Business Case for Web Performance** |
| 100 | Making Performance a Business Priority |
|  | **Module-04: Testing** |
|  | **Introduction to Cross-Browser Testing** |
| 101 | What is Cross-Browser Testing? |
| 102 | Why Do Cross-Browser Issues Occur? |
| 103 | Workflows for Cross-Browser Testing |
| 104 | Reporting Bugs |
| 105 | Summary |
|  | **Strategies for Carrying Out Testing** |
| 106 | Choosing Which Browser and Device to Test |
| 107 | What are You Going to Test? |
| 108 | Putting Together a Testing Lab |
| 109 | User Testing |
| 110 | Summary |
|  | **Handling Common HTML and CSS Problems** |
| 111 | The Trouble with HTML and CSS |
| 112 | First Things First: Fixing General Problems |
| 113 | Common Cross Browser Problems |
| 114 | Finding Help |
| 115 | Summary |
|  | **Implementing Feature Detection** |
| 116 | The Concept of Feature Detection |
| 117 | Writing Your Own Feature Detection Tests |
| 118 | Summary |
|  | **Introduction to Automated Testing** |
| 119 | Automation Makes Things Easy |
| 120 | Using a Task Runner to Automate Testing Tools |
| 121 | Using Commercial Testing Services to Speed Up Browser Testign |
| 122 | Summary |
|  | **Setting Up Your Own Test Automation Environment** |
| 123 | Selenium |
| 124 | Testing in Multiple Browsers at Once |
| 125 | WebDriver Syntax Crash Course |
| 126 | Test Best Practices |
| 127 | Running Remote Tests |
| 128 | Integrating Selenium with CI Tools |
| 129 | Summary |
|  | **Module-05: How to Solve Common Problems (Further Resources)** |
|  | **Solve Common HTML Problems** |
| 130 | Solve Common HTML Problems |
| 131 | Uncommon or Advanced Problems |
| 132 | See Also |
|  | **Solve Common CSS Problems** |
| 133 | Styling Boxes |
| 134 | CSS and Text |
| 135 | CSS Layout |
|  | **Solve Common JavaScript Problems** |
| 136 | Common Beginner’s Mistakes |
| 137 | Basic Definitions |
| 138 | Basic Use Cases |
| 139 | Intermediate Use Cases |
| 140 | **Web Mechanics** |
| 141 | **Tools and Setup** |
| 142 | **Design and Accessibility** |

# **MDN Reference**

|  |  |
| --- | --- |
|  | **Reference** |
|  | **Web Technology for Developers** |
|  | Documentation for Web Developers |
|  | Web Technology References |
|  | Developer Tools Documentation |
|  | **References** |
|  | * Developer Guides |
|  | * Tutorials |
|  | **References** |
|  | * HTML |
|  | * CSS: Cascading Style Sheets |
|  | * JavaScript |
|  | **References** |
|  | * MathML |
|  | * Event Reference |
|  | * SVG |
|  | * XML: Extensible Markup Language |
|  | * XPath |
|  | * XSLT: Extensible Stylesheet Language Transformations |
|  | * EXSLT |
|  | **References** |
|  | * Accessibility |
|  | * Web Performance |
|  | * HTTP |
|  | **References** |
|  | * URIs |
|  | * OpenSearch Description Format |
|  | * Privacy on The Web |
|  | * Security on The Web |
|  | * Progressive Web Apps |
|  | * Web App Manifests |
|  | * Web Media Technologies |
|  | * WebDriver |

# **W3Schools: Database**

|  |  |
| --- | --- |
|  | **W3Schools Database** |
|  | **Module-01: SQL** |
|  | **SQL Query** |
|  | Home and Intro |
|  | Syntax |
|  | Select |
|  | Select Distinct |
|  | Where |
|  | Order By |
|  | And |
|  | Or |
|  | Not |
|  | Insert Into |
|  | Null Values |
|  | Update |
|  | Delete |
|  | Select Top |
|  | Aggregate Functions |
|  | Min and Max |
|  | Count |
|  | Sum |
|  | Avg |
|  | Like |
|  | Wildcards |
|  | In |
|  | Between |
|  | Aliases |
|  | Joins |
|  | Inner Join |
|  | Left Join |
|  | Right Join |
|  | Full Join |
|  | Self-Join |
|  | Union |
|  | Group By |
|  | Having |
|  | Exists |
|  | Any, All |
|  | Select Into |
|  | Insert Into Select |
|  | Case |
|  | Null Functions |
|  | Stored Procedures |
|  | Comments |
|  | Operators |
|  | **SQL Database** |
|  | Create DB |
|  | Drop DB |
|  | Backup DB |
|  | Create Table |
|  | Drop Table |
|  | Alter Table |
|  | Constraints |
|  | Not Null |
|  | Unique |
|  | Primary Key |
|  | Foreign Key |
|  | Check |
|  | Default |
|  | Index |
|  | Auto Increment |
|  | Dates |
|  | Views |
|  | Injection |
|  | Hosting |
|  | Data Types |
|  | **SQL References** |
|  | SQL Keyworks |
|  | MySQL Functions |
|  | **Module-02: MySQL** |
|  | **MySQL Tutorial** |
|  | Home |
|  | Intro |
|  | RDBMS |
|  | **MySQL SQL** |
|  | SQL |
|  | SELECT |
|  | WHERE |
|  | AND, OR, NOT |
|  | ORDER BY |
|  | INSERT INTO |
|  | NULL Values |
|  | UPDATE |
|  | DELETE |
|  | LIMIT |
|  | MIN and MAX |
|  | COUNT, AVG, SUM |
|  | LIKE |
|  | Wildcards |
|  | IN |
|  | BETWEEN |
|  | Aliases |
|  | Joins |
|  | INNER JOIN |
|  | LEFT JOIN |
|  | RIGHT JOIN |
|  | CROSS JOIN |
|  | Self-Join |
|  | UNION |
|  | GROUP BY |
|  | HAVING |
|  | EXISTS |
|  | ANY, ALL |
|  | INSERT SELECT |
|  | Case |
|  | Null Functions |
|  | Comments |
|  | Operators |
|  | **MySQL Database** |
|  | Create DB |
|  | Drop DB |
|  | Create Table |
|  | Drop Table |
|  | Alter Table |
|  | Constraints |
|  | Not Null |
|  | Unique |
|  | Primary Key |
|  | Foreign Key |
|  | Check |
|  | Default |
|  | Create Index |
|  | Auto Increment |
|  | Dates |
|  | Views |
|  | **MySQL Reverences** |
|  | Data Types |
|  | Functions |
|  | **W3Schools Database** |
|  | **Module-03: MongoDB** |
|  | **MongoDb Tutorial** |
|  | Home |
|  | Get Started |
|  | Query API |
|  | Create DB |
|  | Collection |
|  | Insert |
|  | Find |
|  | Update |
|  | Delete |
|  | Query Operators |
|  | Update Operators |
|  | Aggregations |
|  | Indexing/Search |
|  | Validation |
|  | Data API |
|  | Drivers |
|  | Node.js Driver |
|  | Charts |

# **Database Design**

* Beginning Database Design Solution, Rod Stephens

|  |  |
| --- | --- |
|  | **Beginning Database Design Solution** |
|  | **Part-01: Introduction to Database and Database Design** |
|  | **Goals of Effective Database Design** |
|  | Understanding the Importance of Design |
|  | Information Containers |
|  | Strengths and Weakness of Information Containers |
|  | Desirable Database Features |
|  | Summary |
|  | Exercises |
|  | **Database Types** |
|  | Why Bother? |
|  | Flat Files |
|  | Relational Database |
|  | Spreadsheets |
|  | Hierarchical Database |
|  | XML |
|  | Network |
|  | Object |
|  | Object-Relational |
|  | Exotic |
|  | Summary |
|  | Exercises |
|  | **Relational Database Fundamentals** |
|  | Relational Points of View |
|  | Table, Rows, and Columns |
|  | Relations, Attributes, and Tuples |
|  | Keys |
|  | Indexes |
|  | Constraints |
|  | Database Operations |
|  | Summary |
|  | Exercises |
|  | **Part-02: Database Design Process and Techniques** |
|  | **Understanding User Needs** |
|  | Make a Plan |
|  | Bring a List of Questions |
|  | Meet the Customers |
|  | Learn Who’s Who |
|  | Pick the Customers’ Brains |
|  | Walk a Mile in the User’s Shoes |
|  | Study Current Operations |
|  | Brainstorm |
|  | Look to the Future |
|  | Understand the Customers’ Reasoning |
|  | Learning What the Customers Really Need |
|  | Prioritize |
|  | Verify Your Understanding |
|  | Write the Requirements Document |
|  | Make Use Cases |
|  | Decide Feasibility |
|  | Summary |
|  | Exercises |
|  | **Translating User Needs into Data Models** |
|  | What are Data Models? |
|  | User Interface Models |
|  | Semantic Object Models |
|  | Entity-Relationship Models |
|  | Relational Models |
|  | Summary |
|  | Exercises |
|  | **Extracting Business Rules** |
|  | What are Business Rules? |
|  | Identifying Key Business Rules |
|  | Extracting Key Business Rules |
|  | Multi-Tier Applications |
|  | Summary |
|  | Exercises |
|  | **Normalizing Data** |
|  | What is Normalization? |
|  | First Normal Form (1NF) |
|  | Second Normal Form (2NF) |
|  | Third Normal Form (3NF) |
|  | Stopping at Third Normal Form |
|  | Boyce-Codd Normal Form (BCNF) |
|  | Fourth Normal Form (4NF) |
|  | Fifth Normal Form (5NF) |
|  | Domain/Key Normal Form (DKNF) |
|  | Essential Redundancy |
|  | The Best Level of Normalization |
|  | Summary |
|  | Exercises |
|  | **Designing Databases to Support Software Applications** |
|  | Plan Ahead |
|  | Document Everything |
|  | Consider Multi-Tier Architecture |
|  | Convert Domains into Tables |
|  | Keep Tables Focused |
|  | Use Three Kinds of Tables |
|  | Use Naming Conventions |
|  | Allow Some Redundant Data |
|  | Don’t Squeeze in Everything |
|  | Summary |
|  | Exercises |
|  | **Common Design Patterns** |
|  | Associations |
|  | Temporal Data |
|  | Logging and Locking |
|  | Summary |
|  | Exercises |
|  | **Common Design Pitfalls** |
|  | Lack of Preparation |
|  | Poor Documentation |
|  | Poor Naming Standards |
|  | Thinking Tool Small |
|  | Not Planning for Change |
|  | Too Much Normalization |
|  | Insufficient Normalization |
|  | Insufficient Testing |
|  | Performance Anxiety |
|  | Mishmash Tables |
|  | Not Enforcing Constraints |
|  | Obsession with IDs |
|  | Not Defining Natural Keys |
|  | Summary |
|  | Exercises |
|  | **Part-03: A Detailed Case Study** |
|  | **User Needs and Requirements** |
|  | Meet the Customers |
|  | Pick the Customers’ Brains |
|  | Write Use Cases |
|  | Write the Requirements Document |
|  | Demand Feedback |
|  | Summary |
|  | Exercises |
|  | **Building a Data Model** |
|  | Semantic Object Modeling |
|  | Entity-Relationship Modeling |
|  | Relational Modeling |
|  | Putting It All Together |
|  | Summary |
|  | Exercises |
|  | **Extracting Business Rules** |
|  | Identifying Business Rules |
|  | Drawing a New Relational Model |
|  | Summary |
|  | Exercise |
|  | **Normalization and Refinement** |
|  | Improving Flexibility |
|  | Verifying First Normal Form |
|  | Verifying Second Normal Form |
|  | Verifying Second Normal Form |
|  | Verifying Third Normal Form |
|  | Summary |
|  | Exercises |
|  | **Part-04: Implementing Databases** |
|  | **Microsoft Access** |
|  | Understanding Access |
|  | Getting Started |
|  | Defining Relationships |
|  | Creating Field Constraints |
|  | Creating Table Constraints |
|  | Crating Queries |
|  | **MySQL** |
|  | Installing MySQL |
|  | Using MySQL Command Line Client |
|  | Executing SQL Scripts |
|  | Using MySQL Query Browser |
|  | Using MySQL Workbench |
|  | Summary |
|  | Exercises |
|  | **Part-05: Advanced Topics** |
|  | **Introduction to SQL** |
|  | Background |
|  | Finding More Information |
|  | Standards |
|  | Basic Syntax |
|  | Command Overview |
|  | CREATE TABLE |
|  | CREATE INDEX |
|  | DROP |
|  | INSERT |
|  | SELECT |
|  | UPDATE |
|  | DELETE |
|  | Summary |
|  | Exercises |
|  | **Building Databases with SQL Scripts** |
|  | Why Bother with Scripts? |
|  | Script Categories |
|  | Ordering SQL Commands |
|  | Summary |
|  | Exercises |
|  | **Database Maintenance** |
|  | Backups |
|  | Data Warehousing |
|  | Repairing the Database |
|  | Compacting the Database |
|  | Performance Tuning |
|  | Summary |
|  | Exercises |
|  | **Database Security** |
|  | The Right Level of Security |
|  | Passwords |
|  | Privileges |
|  | Initial Configuration and Privileges |
|  | Too Much Security |
|  | Physical Security |
|  | Summary |
|  | Exercises |

# **Software Requirements**

* Requirements Engineering for Software and Systems, Phillip A. Laplante

|  |  |
| --- | --- |
|  | **Requirements Engineering for Software and Systems** |
|  | **Introduction to Requirements Engineering** |
|  | Motivation |
|  | What is Requirements Engineering? |
|  | You Probably Don’t Do Enough Requirement Engineering |
|  | What are Requirements? |
|  | Requirements Engineering Activities |
|  | Bodies of Knowledge |
|  | The Requirements Engineer |
|  | Requirements Engineer Roles |
|  | Role of the Customer |
|  | Problems with Traditional Requirements Engineering |
|  | Difficulties in Enveloping System Behavior |
|  | Exercises |
|  | References |
|  | **Preparing for Requirements Elicitation** |
|  | Production Mission Statement |
|  | Encounter with a Customer |
|  | Identifying the System Boundaries |
|  | Stakeholders |
|  | Customer Wants and Needs |
|  | Why Do Customers Change Their Minds? |
|  | Stakeholder Prioritization |
|  | Communicating with Customers and Other Stakeholders |
|  | Stakeholder Negotiations |
|  | Uncovering Stakeholder Goals |
|  | Exercises |
|  | References |
|  | **Requirements Elicitation** |
|  | Introduction |
|  | Preparing for Requirements Elicitation |
|  | Elicitation Techniques Survey – Topics 6 |
|  | Elicitation Techniques Survey – Topics 6 |
|  | Elicitation Techniques Survey – Topics 6 |
|  | Elicitation Techniques Survey – Topics 4 |
|  | Eliciting Nonfunctional Requirements |
|  | Elicitation Summary |
|  | Eliciting Hazards |
|  | Exercise |
|  | References |
|  | **Writing the Requirements Document** |
|  | Requirements Agreement and Analysis |
|  | Requirements Representation |
|  | ISO/IEC/IEEE Standard 29148 |
|  | UML/SysML |
|  | The Requirement Document – Topics ½ |
|  | The Requirement Document – Topics ½ |
|  | Behavioral Specifications |
|  | Best Practices and Recommendations |
|  | Exercises |
|  | Reference |
|  | **Requirements Risk Management** |
|  | What is Requirements Risk Management? |
|  | Requirements Validation and Verification – Topics ½ |
|  | Requirements Validation and Verification – Topics ½ |
|  | Standards for V&V – Topics ½ |
|  | Standards for V&V – Topics ½ |
|  | NASA Requirements Testing – Topics ½ |
|  | NASA Requirements Testing – Topics ½ |
|  | Exercises |
|  | Reference |
|  | **Formal Methods** |
|  | Motivation |
|  | What Are Formal Methods? |
|  | Examples – Topics 1/3 |
|  | Examples – Topics 1/3 |
|  | Examples – Topics 1/3 |
|  | Objections, Myths, and Limitations |
|  | Bowen and Hinchey’s Advice |
|  | Exercises |
|  | Reference |
|  | **Requirements Specification and Agile Methodologies** |
|  | Introduction to Agile Methodologies |
|  | Extreme Programming |
|  | Scrum |
|  | Requirements Engineering for Agile Methodologies – Topics ½ |
|  | Requirements Engineering for Agile Methodologies – Topics ½ |
|  | Writing User Stories |
|  | Agile Requirements Engineering |
|  | Challenges for Requirements Engineering in Agile Methodologies |
|  | Exercises |
|  | Reference |
|  | **Tool Support for Requirements Engineering** |
|  | Introduction |
|  | Traceability Support |
|  | Requirements Management Tools |
|  | Open-Source Requirements Engineering Tools |
|  | Requirements Engineering Tool Best Practices |
|  | Elicitation Support Technologies |
|  | Requirements Metrics |
|  | Exercises |
|  | References |
|  | **Requirements Management** |
|  | Introduction |
|  | Configuration Management and Control |
|  | Reconciling Differences |
|  | Expectation Revisited: Pascal’s Wager |
|  | Global Requirements Management |
|  | Anti-patterns in Requirements Management – Topics ½ |
|  | Anti-patterns in Requirements Management – Topics ½ |
|  | Other Paradigms for Requirements Management |
|  | Standards for Requirements Management |
|  | Exercises |
|  | References |
|  | **Value Engineering of Requirements** |
|  | What, Why, When, and How of Value Engineering |
|  | Estimating Using COCOMO and Its Derivatives |
|  | Estimating Using Function Points |
|  | Requirements Feature Cost Justification – Topics ½ |
|  | Requirements Feature Cost Justification – Topics ½ |
|  | Putting It All Together |
|  | Exercises |
|  | References |
|  | **Appendix** |
|  | Software Requirements Specification for a Smart Home |
|  | Software Requirements for a Wastewater Pumping Station Web Well Control System |
|  | Unified Modeling Language (UML) |
|  | User Stories |
|  | Use Case |
|  | IBM DOORS Requirements Management Tool |

# **Software Metrics**

* Applied Software Measurement – Global Analysis of Productivity and Quality, Capers Jones

|  |  |
| --- | --- |
|  | **Applied Software Measurement** |
|  | **Introduction** |
|  | Applied Software Measurement |
|  | Planning and Estimation |
|  | Management and Technical Staffs |
|  | Organization Structures |
|  | Methodologies and Tools |
|  | The Office Environment |
|  | Reusability |
|  | The Essential Aspects of Applied Software Measurement |
|  | What Do Companies Measure? |
|  | Benchmarks and Industry Measures |
|  | Measurement and The Software Life Cycle |
|  | The Structure of a Full Applied Software Measurement System |
|  | The Sociology of Software Measurement |
|  | The Sociology of Data Confidentiality |
|  | The Sociology of Using Data for Staff Performance Tragets |
|  | The Sociology of Measuring One-Person Projects |
|  | The Sociology of MIS vs. Systems Software |
|  | Justifying and Building an Applied Software Measurement Function |
|  | Applied Software Measurement and Future Progress |
|  | Suggested Readings |
|  | Additional Readings On Software Measurement and Metrics |
|  | **The History and Evolution of Software Metrics** |
|  | Evolution of the Software Industry and Evolution of Softwre Measuremetns |
|  | The Cons of Counting Function Point Metrics |
|  | The Paradox of Reversed Productivity of High-Level Languages |
|  | The Varieties of Functional Metrics Circa 2008 |
|  | Variations in Application Size and Productivity Rates |
|  | Future Technical Developments in Functional Metrics |
|  | Summary of and Conclusion About Functional Metrics |
|  | Software Measures and Metrics Not Based On Function Points |
|  | Suggested Readings on Measures and Metrics |
|  | **United Stated Averages for Software Productivity and Quality** |
|  | Sources of Possible Errors in the Data |
|  | Significant Software Technology Changes Between 1990 and 2008 |
|  | Changes in the Structure, Format, and Contents of the Third Edition |
|  | Variations in Software Development Practices Among Seven Sub-Industries |
|  | Ranges, Averages, and Variances in Software Productivity |
|  | The Impact of Technology ON Software Productivity and Quality Levels |
|  | Technology Warnings and Counter Indications |
|  | Using Function Point Metrics to Set “Best in Class” Targets |
|  | **The Mechanics of Measurement: Building a Baseline** |
|  | Software Assessments |
|  | Software Baselines |
|  | Software Benchmarks |
|  | What a Baseline Analysis Covers |
|  | Developing or Acquiring a Baseline Data Collection Instrument |
|  | Administering the Data Collection Questionnaire |
|  | Analysis and Aggregation of the Baseline Data |
|  | Suggested Readings |
|  | Additional Readings |
|  | **Measuring Software Quality and User Satisfaction** |
|  | New Quality Information Since the Earlier Editions |
|  | Quality Control and International Competition |
|  | Defining Quality for Measurement and Estimation |
|  | Five Steps to Software Quality Control |
|  | Software Quality Control in the United States |
|  | Measuring Software Defect Removal |
|  | Measuring Defect Removal Efficiency |
|  | Finding and Eliminating Error-Prone Modules |
|  | Using Metrics to Evaluate Test-Case Coverage |
|  | Using Metrics for Reliability Prediction |
|  | Measuring the Costs of Defect Removal |
|  | Evaluating Defect Prevention Methods |
|  | Measuring Customer-Reported Defects |
|  | Measuring Invalid Defects, Duplicate Defects, and Special Case |
|  | Measuring User Satisfaction |
|  | Combining User Satisfaction and Defect Data |
|  | Summary and Conclusions |
|  | Reading List |
|  | Suggested Readings |
|  | Additional References on Software Quality and Quality Measurements |
|  | **Measurements, Metrics, and Industry Leadership** |
|  | What Do Companies Measure? |
|  | Measures and Metrics of Industry Leaders |
|  | Measures, Metrics, and Innovation |
|  | Measurements, Metrics, and Outsource Litigation |
|  | Measurements, Metrics, and Behavioral Changes |
|  | Topics Outside the Scope of Current Measurements |
|  | Cautions Against Simplistic and Hazardous Measures and Metrics |
|  | Commercial Software Measurement Tools |
|  | Summary and Conclusions |
|  | Suggested Readings on Measurement and Metrics |
|  | **Summary of Problems in Software Measurement** |
|  | Synthetic vs. Natural Metrics |
|  | Ambiguity in Defining the Nature, Scope, Class, and Type of Software |
|  | Ambiguity in Defining and Measuring the Activities and Tasks of Software Projects |
|  | False Advertising and Fraudulent Productivity Claims |
|  | The Absence of Project Demographic and Occupation Group Measurement |
|  | Ambiguity in the Span of Control and Organizational Measurements |
|  | The Missing Link of Measurement: When Do Projects Start? |
|  | Ambiguity in Measuring Milestones, Schedules, Overlap, and Schedule Slippage |
|  | Problems with Overlapping Activities |
|  | Leakage from Software Project Resource Tracking Data |
|  | Ambiguity in Standard Time Metrics |
|  | Inadequate Undergraduate and Graduate Training in Software Measurement and Metrics |
|  | Inadequate Standards for Software Measurement |
|  | Lack of Standardization of “Lines of Source Code” Metrics |
|  | The Hazards and Problems of Ratios and Percentages |
|  | Ambiguity in Measuring Development of Delivery Productivity |
|  | Ambiguity in Measuring Complexity |
|  | Ambiguity in Functional Metrics |
|  | Ambiguity in Quality Metrics |
|  | Ambiguity with the Defects Per KLOC Metric |
|  | Ambiguity with the Cost per Defect Metric |
|  | Failure to Measure Defect Potentials and Defect Removal Efficiency |
|  | The Problems of Measuring the Impact of “Soft” Factors |
|  | Problems in Measuring Software Value |
|  | Lack of Effective Measurement and Metrics Automation |
|  | Social and Political Resistance to Software Measurements |
|  | Ambiguity in Software Measurement and Metrics Terminology |
|  | Failure to Use Metrics for Establishing Goals and Targets |
|  | Summary and Conclusions |
|  | Suggested Readings |
|  | Additional References on Software Measurements |
|  | **Appendix: Rules for Counting Procedural Source Code** |
|  | Project Source Code Counting Rules |
|  | General Rules for Counting Code Within Applications |
|  | Examples of the SPR Source Code Counting Rules |
|  | Software Productivity Research COBOL-Counting Rules |

# **Statistics**

* Statistics (A-Level), Sophie Goldie

|  |  |
| --- | --- |
|  | **Applied Software Measurement** |
|  | **S1 Statistics 1** |
|  | **Exploring Data** |
|  | Looking at the Data |
|  | Stem-and-Leaf Diagrams |
|  | Categorical or Qualitative Data |
|  | Numerical or Quantitative Data |
|  | Measures of Central Tendency |
|  | Frequency Distributions |
|  | Grouped Data |
|  | Measures of Spread (Variation) |
|  | Working with an Assumed Mean |
|  | **Representing and Interpreting Data** |
|  | Histograms |
|  | Measures of Central Tendency and of Spread Using Quartiles |
|  | Cumulative Frequency Curves |
|  | **Probability** |
|  | Measuring Probability |
|  | Estimating Probability |
|  | Expectation |
|  | The Probability of Either One Event or Another |
|  | Independent and Dependent Events |
|  | Conditional Probability |
|  | **Discrete Random Variables** |
|  | Discrete Random Variables |
|  | Expectation and Variance |
|  | **Permutations and Combinations** |
|  | Factorials |
|  | Permutations |
|  | Combinations |
|  | The Binomial Coefficients |
|  | Using Binomial Coefficients to Calculate Probabilities |
|  | **The Binomial Distribution** |
|  | The Binomial Distribution |
|  | The Expectation and Variance of B(n, p) |
|  | Using the Binomial Distribution |
|  | **The Normal Distribution** |
|  | Using Normal Distribution Tables |
|  | The Normal Curve |
|  | Modeling Discrete Situations |
|  | Using the Normal Distribution as an Approximation for the Binomial Distribution |
|  | **S2 Statistics 2** |
|  | **Hypothesis Testing Using the Binomial Distribution** |
|  | Defining Terms |
|  | Hypothesis Testing checklist |
|  | Choosing the Significance Level |
|  | Critical Values and Critical (rejection) Regions |
|  | One-Tail and Two-Tail Tests |
|  | Type 1 and Type 2 Errors |
|  | **The Poisson Distribution** |
|  | The Poisson Distribution |
|  | Modelling with a Poisson Distribution |
|  | The Sum of Two or More Poisson Distributions |
|  | The Poisson Approximation to the Binomial Distribution |
|  | Using the Normal Distribution as an Approximation for the Poisson Distribution |
|  | **Continuous Random Variables** |
|  | Probability Density Function |
|  | Mean and Variance |
|  | The Median |
|  | The Mode |
|  | The Uniform (rectangular) Distribution |
|  | **Linear Combinations of Random Variables** |
|  | The Expectation (Mean) of a Function of X, E(g[X]) |
|  | Expectation: Algebraic Results |
|  | The Sums and Differences of Independent Random Variables |
|  | More Than Two Independent Random Variables |
|  | **Sampling** |
|  | Terms and Notation |
|  | Sampling |
|  | Sampling Techniques |
|  | **Hypothesis Testing and Confidence Intervals Using the Normal Distribution** |
|  | Interpreting Sample Data Using the Normal Distribution |
|  | The Central Limit Theorem |
|  | Confidence Intervals |
|  | How Large a Sample Do You Need? |
|  | Confidence Intervals for a Proportion |