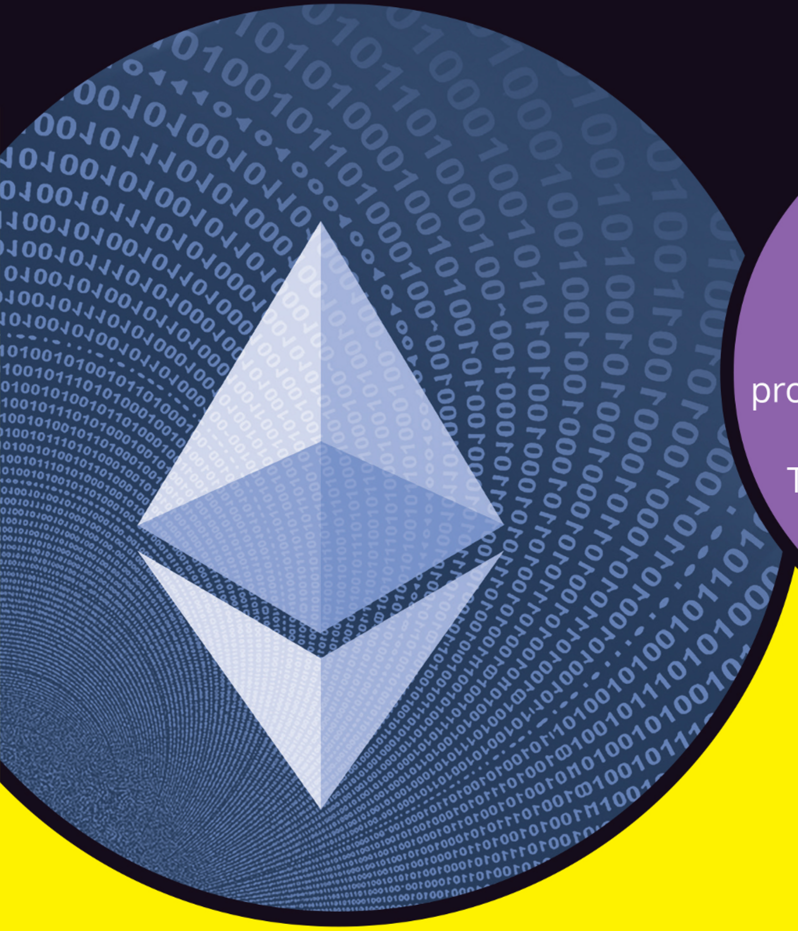
LEARNING MADE EASY

# Ethereum™

for dummies®

A Wiley Brand



Learn how Ethereum is changing business

Insight for investors, programmers, and security pros

The tools you need to build Ethereum applications

Michael G. Solomon

# Ethereum™

by Michael G. Solomon

for dummies®
A Wiley Brand

## Ethereum™ For Dummies®

Published by: **John Wiley & Sons, Inc.,** 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Published simultaneously in Canada

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit https://hub.wiley.com/community/support/dummies.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2019936125

ISBN 978-1-119-47412-8 (pbk); ISBN 978-1-119-47411-1 (ebk); ISBN 978-1-119-47406-7 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

# Contents at a Glance

**Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

**Part 1: Getting to Know Blockchain and Ethereum** . . . . . . . . . 5

CHAPTER 1: Introducing Ethereum . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

CHAPTER 2: Learning about Blockchain . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

CHAPTER 3: Exploring Use Cases for Ethereum . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

**Part 2: Setting Up Your Ethereum Development Environment** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

CHAPTER 4: Examining the Ethereum Ecosystem and Development Lifecycle . . . . . 59

CHAPTER 5: Getting and Configuring Ethereum Development Tools . . . . . . . . . . . . 77

CHAPTER 6: Establishing an Ethereum Wallet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 95

**Part 3: Building Ethereum Distributed Blockchain Apps** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107

CHAPTER 7: Building Your First Ethereum Apps . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 109

CHAPTER 8: Learning about Smart Contracts . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 125

CHAPTER 9: Writing Your Own Smart Contracts with Solidity. . . . . . . . . . . . . . . . . . . 147

**Part 4: Testing and Deploying Ethereum Apps** . . . . . . . . . . . . . 173

CHAPTER 10: Testing Ethereum Apps . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 175

CHAPTER 11: Deploying and Maintaining Ethereum Apps . . . . . . . . . . . . . . . . . . . . . 191

CHAPTER 12: Integrating Non-Blockchain Apps with Ethereum. . . . . . . . . . . . . . . . . . 205

**Part 5: The Part of Tens** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 219

CHAPTER 13: Ten Free Ethereum Resources . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 221

CHAPTER 14: Ten Design Principles for Distributed Blockchain Apps . . . . . . . . . . . . . 229

CHAPTER 15: Top Ten Ethereum Projects. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 239

**Index** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 247

# Table of Contents

INTRODUCTION . . . . . . . . . . 1
About This Book. . . . . . . . . . 1
Foolish Assumptions. . . . . . . . . . 2
Icons Used in This Book . . . . . . . . . . 2
Beyond the Book . . . . . . . . . . 2
Where to Go from Here . . . . . . . . . . 3

PART 1: GETTING TO KNOW BLOCKCHAIN AND ETHEREUM . . . . . . . . . . 5

CHAPTER 1: Introducing Ethereum . . . . . . . . . . 7
Describing Blockchain Technology . . . . . . . . . . 8
Introducing Ethereum. . . . . . . . . . 10
Exploring Ethereum's Consensus, Mining, and Smart Contracts . . . . . 11
Buying, Spending, and Trading Ether . . . . . . . . . . 13
Getting Started with DAO and ICO . . . . . . . . . . 16
Exploring the Ethereum Ecosystem . . . . . . . . . . 17
Delving into Development Tools . . . . . . . . . . 18
Building Blockchain Apps . . . . . . . . . . 18

CHAPTER 2: Learning about Blockchain . . . . . . . . . . 21
Exploring Distributed Applications . . . . . . . . . . 22
Digging into distributed processing . . . . . . . . . . 22
Exploring problems with distributed processing . . . . . . . . . . 24
Presenting some solutions to distributed processing problems . . . 27
Examining the Bitcoin Solution to the Distributed Dilemma . . . . . . . . 28
Describing Blockchains. . . . . . . . . . 30
Examining blockchain details. . . . . . . . . . 30
Protecting blockchain visibility . . . . . . . . . . 31
Building Blockchains . . . . . . . . . . 33
Agreeing to add blocks . . . . . . . . . . 33
Making blocks immutable . . . . . . . . . . 34
Reviewing the building process. . . . . . . . . . 35
Keeping all blockchains consistent . . . . . . . . . . 35
Understanding How Blockchains and Databases Store Data Differently. . . . . . . . . . 36
Storing data in a traditional database . . . . . . . . . . 36
Storing data in a blockchain. . . . . . . . . . 38
Effectively Using Blockchains. . . . . . . . . . 39
Transferring value without trust . . . . . . . . . . 39
Reducing transaction costs by eliminating middlemen. . . . . . . . . . 39

Increasing efficiency through direct interaction . . . . . . . . . . . . . . . . 40
Maintaining complete transaction history. . . . . . . . . . . . . . . . . . . . . 40
Increasing resilience through replication. . . . . . . . . . . . . . . . . . . . . 41
Providing transparency. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

CHAPTER 3: **Exploring Use Cases for Ethereum**. . . . . . . . . . . . . . . . . . . 43
Diving Into Ethereum Applications . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
Exploring Financial Services. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
Banking . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
Creating Ethereum escrow applications . . . . . . . . . . . . . . . . . . . . . . 48
Examining ICOs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
Establishing Digital Identity Management . . . . . . . . . . . . . . . . . . . . . . 49
Managing individual and device identities. . . . . . . . . . . . . . . . . . . . 50
Reducing fraud and identity theft . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
Examining the ERC-725 standard and beyond . . . . . . . . . . . . . . . . . 51
Examining Industry Applications. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
Healthcare . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
Energy . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
Supply chain . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
Enabling Effective Governance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
Tax payment. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
Government spending . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
Voting . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
Policy development. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
Notary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56

PART 2: SETTING UP YOUR ETHEREUM DEVELOPMENT ENVIRONMENT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

CHAPTER 4: **Examining the Ethereum Ecosystem and Development Lifecycle** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
Exploring the Ethereum Blockchain Block Structure . . . . . . . . . . . . . . . 60
Describing Smart Contracts. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
Introducing Solidity, the Language of Smart Contracts . . . . . . . . . . . . . 66
Working with the Ethereum Virtual Machine . . . . . . . . . . . . . . . . . . . . 67
Fueling Your Code with Gas. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68
Surveying Tools for Developing, Testing, and Deploying Ethereum Apps . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
Ethereum blockchain client . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
Development and testing blockchain. . . . . . . . . . . . . . . . . . . . . . . . . 71
Compiler and testing framework . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
Source code editor/IDE. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
Describing the Ethereum Development Lifecycle. . . . . . . . . . . . . . . . . . 73
Introducing Smart Contract Development Tools . . . . . . . . . . . . . . . . . . 74

CHAPTER 5: **Getting and Configuring Ethereum Development Tools** . . . 77
Examining Why Multiple Ethereum Development Tools Are Available . . . 78
Downloading, Installing, and Configuring All the Pieces . . . 79
Installing the blockchain client . . . 79
Installing the test blockchain . . . 83
Installing the testing environment . . . 86
Installing the IDE . . . 91

CHAPTER 6: **Establishing an Ethereum Wallet** . . . 95
Unlocking the Secrets of an Ethereum Wallet . . . 96
Examining the Types of Ethereum Wallets . . . 96
Sorting out software wallets . . . 97
Handling hardware wallets . . . 99
Perusing paper wallets . . . 99
Choosing an Ethereum Wallet . . . 100
Software wallets . . . 100
Hardware wallets . . . 102
Paper wallets . . . 103
Installing MetaMask, an Ethereum Wallet . . . 104

PART 3: BUILDING ETHEREUM DISTRIBUTED BLOCKCHAIN APPS . . . 107

CHAPTER 7: **Building Your First Ethereum Apps** . . . 109
Validating Your Ethereum Development Environment . . . 110
Creating a Truffle project . . . 110
Editing the Truffle config file . . . 111
Exploring the Ganache Test Environment . . . 113
Designing Simple Smart Contracts . . . 115
Coding Your First Smart Contract . . . 116
Running Your First Smart Contract . . . 118
Writing your code . . . 118
Compiling your code . . . 119
Deploying your code . . . 120
Invoking your code's functions . . . 122
Paying as You Go . . . 124

CHAPTER 8: **Learning about Smart Contracts** . . . 125
Introducing Supply Chain and Common Challenges . . . 126
Describing supply chain . . . 126
Explaining difficulties when implementing a supply chain . . . 127

Examining How Blockchain Can Help Resolve Supply Chain Problems. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .128
Describing a Blockchain Supply Chain Solution . . . . . . . . . . . . . . . . . . .129
Paying for supply chain services . . . . . . . . . . . . . . . . . . . . . . . . . . . .129
Managing assets on the supply chain . . . . . . . . . . . . . . . . . . . . . . .130
Digging into Solidity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .132
Describing Basic Smart Contract Syntax . . . . . . . . . . . . . . . . . . . . . . .133
Declaring valid compiler version. . . . . . . . . . . . . . . . . . . . . . . . . . . .134
Commenting your code . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .134
Importing external code. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .135
Defining your smart contracts. . . . . . . . . . . . . . . . . . . . . . . . . . . . .135
Handling Data in Solidity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .136
Learning about Computation and Gas. . . . . . . . . . . . . . . . . . . . . . . . . .140
Exploring Access Modes and Visibility of Smart Contract Functions and Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .142
Controlling Execution Flow. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .144
Handling Errors and Exceptions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .145

CHAPTER 9: **Writing Your Own Smart Contracts with Solidity** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 147
Reviewing Supply Chain Design Specification. . . . . . . . . . . . . . . . . . . .148
Payment token smart contract . . . . . . . . . . . . . . . . . . . . . . . . . . . . .149
Supply chain smart contract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .150
Creating New Smart Contracts . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .151
ERC-20 token interface . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .153
ERC-20 token smart contract. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .154
Supply chain smart contract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .155
Coding Primary Functions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .157
ERC-20 token functions. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .157
Supply chain functions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .160
Using Events. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .163
Defining events . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .165
Triggering events . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .166
Introducing Ownership. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .168
Designing for Security. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .170
Implementing Minimal Functionality . . . . . . . . . . . . . . . . . . . . . . . . . .171

PART 4: TESTING AND DEPLOYING ETHEREUM APPS . . . . 173

CHAPTER 10: **Testing Ethereum Apps**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 175
Understanding Ethereum dApp Testing . . . . . . . . . . . . . . . . . . . . . . .176
Writing tests from the beginning . . . . . . . . . . . . . . . . . . . . . . . . . . .176
Choosing the right test blockchain . . . . . . . . . . . . . . . . . . . . . . . . . .176
Learning the steps in the testing lifecycle . . . . . . . . . . . . . . . . . . . . .177
Testing for software quality . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .177

Deploying a dApp to a Test Ethereum Blockchain . . . . . . . . . . . . . . . .178
Telling Truffle to use the Ganache blockchain . . . . . . . . . . . . . . . .178
Deploying your code to the Ganache blockchain . . . . . . . . . . . . . .180
Writing Tests for Ethereum dApps . . . . . . . . . . . . . . . . . . . . . . . . . . .181
Testing using the command line . . . . . . . . . . . . . . . . . . . . . . . . . .181
Writing test cases in JavaScript . . . . . . . . . . . . . . . . . . . . . . . . . .185
Logging and Handling Errors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .187
Handling errors in Solidity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .188
Logging activity in smart contracts . . . . . . . . . . . . . . . . . . . . . . . .189
Fixing Bugs in a dApp . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .190

CHAPTER 11: **Deploying and Maintaining Ethereum Apps** . . . . . . . 191
Test Blockchain Options versus Live Blockchains. . . . . . . . . . . . . . . . .192
Testing with the Ganache blockchain. . . . . . . . . . . . . . . . . . . . . . . .192
Deploying your code to other test blockchains. . . . . . . . . . . . . . . .193
Anticipating Differences in Live Environments . . . . . . . . . . . . . . . . . . .195
Preparing Your Configuration for Deploying to Different Networks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .196
Deploying a dApp . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .198
Getting enough ether . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .199
Compiling your code . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .203
Deploying your code . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .203

CHAPTER 12: **Integrating Non-Blockchain Apps with Ethereum** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
Comparing Blockchain and Database Storage . . . . . . . . . . . . . . . . . . .206
Locating control . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .207
Imposing data format . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .207
Updating data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .207
Optimizing performance . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .208
Protecting confidentiality . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .208
Paying for storage . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .208
Providing integrity and transparency . . . . . . . . . . . . . . . . . . . . . . .209
Protecting resilience . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .209
Contrasting Execution and Flow in Blockchain dApps and Traditional Applications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .210
Designing Goals for Incorporating Blockchain into an Existing Application . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .211
Using existing smart contracts . . . . . . . . . . . . . . . . . . . . . . . . . . . .213
Developing your own smart contracts . . . . . . . . . . . . . . . . . . . . . . .213
Identifying Interface Data and Transaction Requirements . . . . . . . . .214
Creating or Modifying Contracts to Provide Data Interface . . . . . . . .215
Testing Integrated dApps . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .215
Deploying Integrated dApps . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .216

PART 5: THE PART OF TENS . . . . . . . . . . 219

CHAPTER 13: Ten Free Ethereum Resources . . . . . . . . . . 221
Exploring Alternative Ethereum Development Frameworks . . . . . . . 222
Managing you development with Populus . . . . . . . 222
Exploring Ethereum blockchain containers with Cliquebait . . . . . 222
Selecting a Free Integrated Development Environment . . . . . . . 223
Developing Solidity code with Atom . . . . . . . 223
Going online with Remix . . . . . . . 224
Keeping it simple with EthFiddle . . . . . . . 224
Exploring Ethereum Clients and APIs . . . . . . . 226
Swapping your Ethereum client to Parity . . . . . . . 226
Interacting with Ethereum by using Web3.js . . . . . . . 226
Focusing on Wallets and Security . . . . . . . 227
Protecting your crypto-assets in Mist . . . . . . . 227
Securing your dApps with OpenZeppelin . . . . . . . 228
Learning More About Developing Ethereum dApps . . . . . . . 228

CHAPTER 14: Ten Design Principles for Distributed Blockchain Apps . . . . . . . . . . 229
Designing for Trust . . . . . . . 230
Enforcing Consistency . . . . . . . 230
Removing Doubt through Transparency . . . . . . . 232
Providing Feedback, Guidance, and Setting Expectations . . . . . . . 233
Handling Mistakes with Class . . . . . . . 233
Designing Functions that Focus on User Actions, Not Data . . . . . . . 234
Storing Data Based on User Actions, Not Data Structures . . . . . . . 235
Keeping It Simple . . . . . . . 236
Expecting Blockchain Access to Be Expensive . . . . . . . 236
Staying Out of the User's Way . . . . . . . 237

CHAPTER 15: Top Ten Ethereum Projects . . . . . . . . . . 239
Predicting Future Events with Gnosis . . . . . . . 240
Crowdsourcing Event Predictions in Augur . . . . . . . 240
Managing Decentralized Organizations with Aragon . . . . . . . 241
Breeding and Collecting Cryptokitties . . . . . . . 241
Exchanging Tokens with IDEX . . . . . . . 242
Creating Your Digital Identity with uPort . . . . . . . 243
Sharing Your Thoughts on the Blockchain with EtherTweet . . . . . . . 243
Searching for Jobs with EthLance . . . . . . . 244
Using TenX to Pay with Cryptocurrency . . . . . . . 245
Buying and Selling Computing Power with Golem . . . . . . . 246

INDEX . . . . . . . . . . 247

# Introduction

Blockchain technology is one of the most talked about disruptive technologies of the decade, and Ethereum is the most popular blockchain implementation. Blockchain technology holds the promise of making business interactions faster, cheaper, and more trustworthy.

*Ethereum For Dummies* introduces blockchain and Ethereum, covers their effect on today's ways of doing business, and teaches you how to design and develop your own Ethereum decentralized applications. You learn how to set up a development environment and write smart contracts that create and control transactions on the Ethereum blockchain.

## About This Book

Blockchain technology has the potential to change how business operates. The unprecedented opportunities blockchain promises to provide include easy data sharing among large groups, transparency, trusted transactions, and complete historical audit trails. Today, most organizations protect transaction data as a valued asset, but sharable data could change everything. Sharing trusted data with many participants in a business process has the potential of revolutionizing how organizations interact with one another, reducing the need and cost of middlemen and providing unprecedented transparency to business processes.

Staying current and pertinent means becoming part of this emerging blockchain business model. *Ethereum For Dummies* gives you the foundation of blockchain and Ethereum, and teaches you, in clear language, how to design and write your own software for the Ethereum blockchain environment.

## Foolish Assumptions

I don't make many assumptions about your experience with blockchain technology, application programming, or cryptography, but I do assume the following:

- You have a computer and access to the Internet.
- You know the basics of using your computer and the Internet, and how to download and install programs.
- You know how to find files on your computer's disk and how to create folders.
- You're new to blockchain and you aren't an experienced software developer. If you already know how to write software applications, you can skip the sections on programming basics.

## Icons Used in This Book

TIP

The Tip icon marks tips (duh!) and shortcuts that you can use to make learning and using Ethereum and Solidity easier.

REMEMBER

Remember icons mark the information that's especially important to know. To siphon off the most important information in each chapter, just skim through these icons.

TECHNICAL STUFF

The Technical Stuff icon marks information of a highly technical nature that you can normally skip over.

WARNING

The Warning icon tells you to watch out! It marks important information that may save you headaches when writing your own blockchain applications.

## Beyond the Book

In addition to the material in the print or e-book you're reading right now, this product also comes with some access-anywhere goodies on the web. Check out the free cheat sheet for more on Ethereum and Solidity at `www.dummies.com/cheatsheet/ethereumfd`.

You'll find summary information on Ethereum and Solidity tools as well as tips on how to use them effectively. The cheat sheet is a reference to use over and over as you gain experience in developing Ethereum decentralize applications.

In addition, if you'd rather download the code you see in this book instead of typing it, go to `www.dummies.com/go/ethereumfd`. You can download zip files for each of the projects you'll create to develop and test smart contracts.

## Where to Go from Here

The *Dummies* series tells you what you need to know and how to do the things you need to do to get the results you want. Readers don't have to read the entire book to just learn about some topics. For example, if you just want to learn about smart contracts, you can jump right to Chapters 8 and 9. On the other hand, if you need to set up your own development environment, read Part 2, which tells you how to do that with clear, step-by-step instructions.

# 1 Getting to Know Blockchain and Ethereum

## IN THIS PART . . .

Get a big-picture overview of the Ethereum blockchain and how it works.

Discover how blockchain technology addresses distributed application problems.

Explore use cases that are good fits for blockchain technology.

IN THIS CHAPTER

- » Getting to know Ethereum
- » Understanding ether and trading
- » Exploring DAOs and ICOs
- » Setting up an Ethereum development environment
- » Developing blockchain applications

# Chapter 1
# Introducing Ethereum

Blockchain technology is the most disruptive technology introduced in our generation, and Ethereum is by far the most popular blockchain implementation in use today. You can't read many technology articles or blogs without seeing something about how blockchain changes everything. Although some claims seem to be a little far-fetched, blockchain technology really is a game-changer.

Blockchain, which first burst on the scene in 2008, has gained global notoriety for what it has already changed and what is coming. At first, blockchain was all about a new type of electronic currency. But now, partially thanks to Ethereum, blockchain is so much more than a new way to pay for things. It's a new way to think about things. It enables people and businesses to conduct business without many of the obstacles that have existed in trade relations for centuries.

In this book you learn about what blockchain is and why it is viewed as so radical. You discover how powerful Ethereum is in diverse domains and how you can harness its promise and power in your own organizations. If you want to learn what Ethereum is and how it can work for you without having to trudge through hundreds of pages of theory and background, this is the book for you.

# Describing Blockchain Technology

You learn a lot more about blockchain technology in Chapter 2, but before you meet Ethereum, you need to know a little of Ethereum's backstory.

TIP

If you already know what blockchain is, this section will be like watching yet another depiction of why Bruce Wayne became Batman. Feel free to skim it and move on to the next section. There are only so many ways you can kill Thomas and Martha Wayne.

Blockchain technology was introduced to support a new type of digital currency that you can trade in a trustless environment. Traditional currency exchanges require a trusted third party between the parties. Even if a buyer provides coins or bills to a seller at the point of transaction, some government provides the guarantee of the currency's value. There is always a middleman. If the exchange involves a payment card or check, other financial institutions participate to handle the transfer of funds between parties.

In 2008, Satoshi Nakamoto published a paper that changed everything. Nakamoto's paper described a new way to store and distribute data with verifiable integrity among a group of nodes that don't trust one another. You learn more about how Nakamoto's proposal works, and about bitcoin, the cryptocurrency proposed in the paper, in Chapter 2. At this point, the most important takeaway is that this paper showed how to take the requirement for a trusted (and omnipotent) central authority out of the mix. Using this new technology, called blockchain, application developers can create environments in which nodes that do not trust one another can share data that they can trust.

The idea is based on several concepts that are simple to consider but difficult to put into practice. First, data is logically presented as a ledger. The data isn't really stored that way; you can just think of it as a ledger. A ledger is a way of recording data as transactions occur. One interesting feature of this ledger is that you can only add data to it. You can't change anything after you've added it. So, the only two operations you can perform on this ledger are add and read. We refer to the "add only" property as the *immutability property.* In short, blockchains are immutable. As you'll see, immutability is crucial for the technique to work.

Another feature is that data is added to this ledger in blocks. *Blocks* are collections of transactions, each with an owner's address. *Addresses* are the unique IDs of accounts in our ledger system. When there are enough transactions to make a new block, some of the blockchain participants begin a process of adding a new block to the ledger. Each new block is linked to the previous block, making a chain. That's where the term blockchain comes from. A *blockchain* is basically a bunch of blocks where each block is connected to its predecessor.

Then, the entire set of blocks, or the entire blockchain, is shared with other participants. These participants are called *nodes.* These nodes communicate with one another and each stores an exact copy of the blockchain. Many blockchain networks are made up of thousands of nodes, and keeping all of the copies of the blockchain in *sync* (that is, ensuring that every copy is the same) is another revolutionary feature of blockchain technology.

Blockchain technology is built on a *democracy governance mode.* Before any new block is added to the blockchain, a majority of nodes must agree that the new block is valid. All nodes agree to accept the majority decision. That's how the blockchain stays in sync. Nodes essentially vote on all new blocks. Different blockchains use different voting methods, but one of the more common ones requires nodes to solve very hard mathematical puzzles to earn the right to add a new block to the blockchain. As an incentive for doing the hard work, the node that solves the puzzle first gets a reward. The reward encourages nodes to pitch in and help do the hard work of solving verification puzzles.

Part of the puzzle solution involves creating a mathematical hash of the previous block. By storing the previous block's hash in the current block, every node can quickly determine if any block has changed. Each node periodically scans the blockchain to ensure that nothing has changed. This is how nodes can be sure that the blockchain is the same across the entire network. And, because no block can change after it is added to the blockchain, you never have to worry about overwriting data.

Putting it all together, a blockchain makes it possible to share a set of data with many nodes that you don't trust. You can trust the democracy of the network, though. As long as you can trust that more than half of the nodes on the blockchain network are going to be honest, you can trust the blockchain.

The last big advantage to blockchain technology is that you can put rules of operation in blocks on the blockchain as well. These rules are called *smart contracts.* A smart contract is just a program that lives in a blockchain block and governs how data is added to the blockchain. Because all blockchain data is immutable, even the smart contract code is immune from changes. That's how you can exchange currency without a bank. As long as there are rules that dictate how a currency exchange is carried out, transaction data can be recorded on the blockchain and be part of the permanent ledger.

For example, suppose you want to buy a car. You have enough digital currency in your blockchain account to buy the car, and the car owner has the car's title stored in the blockchain. You can offer to buy the car and if the seller accepts your offer, a smart contract handles the transaction. The smart contract would verify that the title is owned by the seller and that you have enough money in your account to make the purchase. If those two requirements are met, the smart contract would transfer the sales amount into the seller's account and transfer the title to your account. Without any middleman, you have purchased a car and paid for it without carrying a wad of cash around.

TIP

Of course, you really purchased a title to a car. Blockchain handles digital assets. You still have to physically get the keys and the car from the seller.

# Introducing Ethereum

*Bitcoin* was the first blockchain technology application. It was revolutionary and defined the first widely used digital currency, called *cryptocurrency*. The *crypto* part of the name refers to the use of cryptographic hashes to ensure the integrity of the blockchain. The shared ledger literally keeps a copy of every cryptocurrency transaction that gets verified by all nodes. Using this approach, bitcoin created a permanent record of every exchange of their cryptocurrency. And, because account owners are identified only by an address, bitcoin has always enjoyed a measure of anonymity.

TECHNICAL STUFF

Although bitcoin addresses aren't linked directly to people, many exchanges have records of identities that are related to addresses. At some point, you have to exchange your cryptocurrency for real currency. That switchover point is where many law enforcement officials focus when they're trying to track down criminals using cryptocurrency.

As bitcoin became more and more popular, researchers began to see more applications for blockchain technology beyond cryptocurrency. In 2013, Vitalik Buterin, the cofounder of *Bitcoin Magazine*, published a whitepaper that proposed a new, more functional blockchain implementation. This new proposal was for the Ethereum blockchain. After gaining interest and attracting technical and financial support, the Ethereum Foundation, a Swiss non-profit organization, was founded and became the developer of Ethereum.

Ethereum wasn't created just to exchange cryptocurrency. In fact, it was designed from the beginning to be different. The core features of Ethereum are the smart contract and ether. *Ether* is the native cryptocurrency that Ethereum supports, although you can create your own tokens to exchange value in many other forms. Smart contracts provide an execution environment that ensures integrity across all nodes. Any code that executes on one node executes the same way on all nodes. This guarantee makes it possible to deploy a wide range of applications across untrusted environments.

The foundational guarantees Ethereum provides support many types of value exchanges without the concern about fraud, censorship, or any involvement by a third party. When you interact with an Ethereum application, you don't have to rely on any intermediary to broker your transactions. You don't need a bank, wholesaler, or transaction broker to provide trust. As a result of Ethereum's *disintermediation,* you can often complete transactions faster, with far lower service fees and without requiring approval from external authorities.

Ethereum is a comprehensive, decentralized application platform that expands the range of capabilities beyond what was possible before blockchain technology. Whereas legacy solutions to data and process sharing required third-party authorities to enforce integrity, Ethereum provides process and data integrity, along with disintermediation. The possibilities are just beginning to be explored.

# Exploring Ethereum's Consensus, Mining, and Smart Contracts

Ethereum provides integrity in the way it implements immutability and smart contracts. Immutability isn't actually a blockchain guarantee. You can change data in any block — even after other blocks are added to the blockchain. However, as soon as you change a block, that block and all subsequent blocks fail integrity checks and your node is out of sync. Instead of saying that the blockchain is immutable, it is more accurate to say that any changes (mutations) to the blockchain are easily and immediately detected.

Ethereum is based on democracy. Each node gets an equal vote. Every time nodes get a new block to add to the blockchain, they validate the block and its transactions, and then vote whether to accept or reject the block. If several different blocks are submitted by different nodes, only one of the blocks can receive votes from a majority. The block that gets more than half of the network node's votes gets to join the blockchain as its newest block.

One of the first problems is to determine when a new block is ready for the blockchain. When too many conflicting blocks are submitted, the voting process slows down. Ethereum makes it hard to add new blocks to keep the number of new block collisions low and to make voting faster. Ethereum uses a consensus protocol called *Proof of Work (PoW),* which sets the rules for validating and adding new blocks. PoW makes add blocks to the blockchain difficult but profitable.

Ethereum defines ether as its cryptocurrency. You can transfer ether between accounts or earn it by doing the hard work of adding blocks to the Ethereum blockchain. The Ethereum PoW mechanism requires that nodes find a number that, when combined with the block's header data, produces a cryptographic hash value that matches the current target, which is a value that is adjusted to keep new block production at a steady rate. Finding a hash value that matches the current target is hard. You have to try on average more than a quadrillion values to find the right one. That's the point. Using a PoW mechanism makes it so hard to submit a block that fewer blocks are submitted, which reduces the number of collisions. The node that finds the right value gets a small ether payment for the effort. This process is called *mining,* and the node that wins the prize is that block's *miner.*

Mining regulates the speed at which new blocks get submitted as candidate blocks, and results in a number that is easy to validate. Finding the right number to solve the puzzle is difficult, but verifying the number is fast and easy. Another interesting aspect of mining is that each block's header contains a hash from the previous block. Ethereum nodes use the hash to easily detect unauthorized block changes. If a block changes, the hash result doesn't match and the block becomes invalid.

Mining is also a way to make money using blockchain technology. Mining has become competitive, and most of today's miners invest in high-performance hardware with multiple GPUs to carry out the complex operations. To keep the mining process fair, Ethereum uses a complexity value that makes the mining process even harder as miners get faster. Adjusting the complexity allows Ethereum to regulate the new block frequency to an average of one new block every 14 seconds.

The glue that holds the Ethereum environment together is the smart contract. Ethereum is much more than just a financial ledger, and smart contracts provide much of its rich functionality. Each Ethereum node runs a copy of the *Ethereum virtual machine (EVM).* The EVM runs smart contract code in a way that guarantees that smart contracts execute the same way on all nodes and produce the same output. Running smart contract code is not optional. Smart contracts execute based on specific rules and cannot be subverted or halted. The EVM smart contract guarantees provide a stable platform for automated transaction processing that you can trust. Smart contracts provide the primary power of the Ethereum environment.

One of the known weaknesses with software is that attackers can sometimes bypass its controls and carry out unintended actions. That type of attack is more difficult in Ethereum, primarily due to its smart contract implementation. Attackers can't directly attack the blockchain and make unauthorized changes because any such changes will be immediately detected. The next most likely attack vector is the smart contract interface to the blockchain data. Ethereum guarantees that smart contract code, which is translated into *bytecode* before it is written to the blockchain, executes on every EVM instance the same way. Also, the EVM determines when code executes and what code executes. Attackers have few opportunities to leverage smart contract code, which makes Ethereum an even more secure environment.

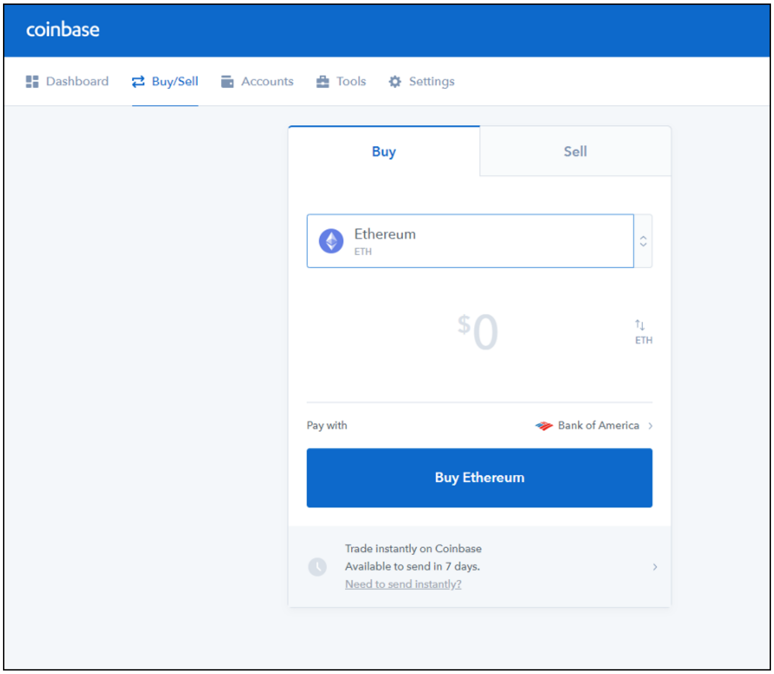
# Buying, Spending, and Trading Ether

Ethereum runs on ether (ETH), its main cryptocurrency. The majority of all existing ether was pre-mined when Ethereum first went live on July 30, 2015. Miners continually create ether, but the amount of mined ether is less than 30 percent of all ether in existence. The lifecycle of Ethereum transactions requires that you first acquire ether to participate in Ethereum. Many exchanges support exchanging legal tender, also called *fiat currency,* for cryptocurrency, including ether. You can navigate to `https://99bitcoins.com/best-ethereum-exchange-review-comparison` for an independent comparison of several popular exchanges.

Before you can interact with the Ethereum blockchain, you need to create at least one account. Creating an Ethereum account is essentially just creating a cryptographic private and public key pair, and generating the associated address, which is based on your public key. The software that handles this process is called an Ethereum *wallet.* You learn about different options for Ethereum wallets in Chapter 6. You can use a wallet provided by an exchange or a standalone wallet. After you create your Ethereum account, you'll need to select an exchange to purchase ether.

After you select an exchange, you set up an exchange account and provide a funding source. Your main funding source is generally a bank account. The most common way to buy ether is to withdraw funds from your bank account and use that money to exchange for ether. Figure 1-1 shows the purchase ether web page for the coinbase.com exchange. Note that the funding source for this account is a Bank of America account.

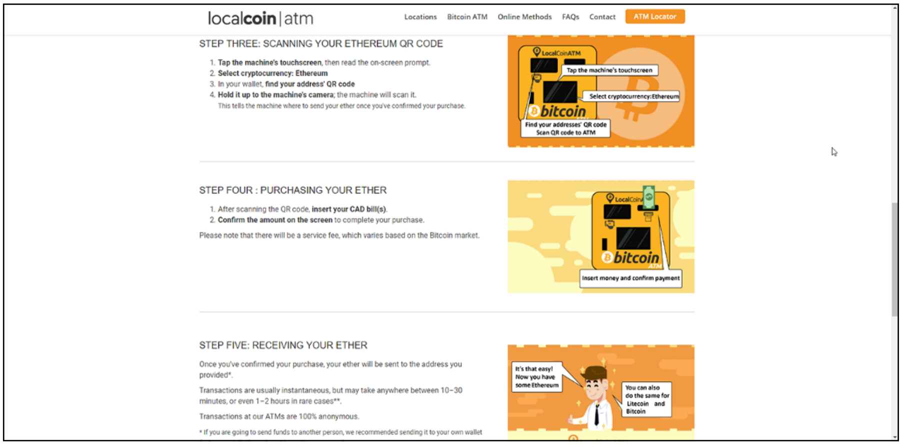
You can also purchase ether using cash. A growing number of cryptocurrency ATMs allow you to exchange cash for different types of cryptocurrency. All you need is the private key you generated using your Ethereum wallet and cash. However, you will pay for this convenience. Cryptocurrency ATMs often use exchange rates that are less favorable than more traditional exchanges. One current service, localcoin ATM, works just like a regular ATM. Navigate to `https://localcoinatm.com` to see where you can find ATMs and how to use them. Figure 1-2 shows several steps in the process of purchasing ether with cash from an ATM.



Pay with Bank of America

Trade instantly on Coinbase
Available to send in 7 days.
Need to send instantly?

**FIGURE 1-1:** Purchase ether using coinbase.com.


localcoin | atm
Locations
Bitcoin ATM
Online Methods
FAQs
Contact
ATM Locator
LocalCoinATM
Tap the machine's touchscreen
Select cryptocurrency:Ethereum
bitcoin
Find your addresses' QR code
Scan QR code to ATM
LocalCoin
bitcoin
Insert money and confirm payment
It's that easy! Now you have some Ethereum
You can also do the same for Litecoin and Bitcoin

STEP THREE: SCANNING YOUR ETHEREUM QR CODE

1. **Tap the machine's touchscreen**, then read the on-screen prompt.
2. **Select cryptocurrency: Ethereum**
3. In your wallet, **find your address' QR code**
4. **Hold it up to the machine's camera**; the machine will scan it.
   This tells the machine where to send your ether once you've confirmed your purchase.

STEP FOUR : PURCHASING YOUR ETHER

1. After scanning the QR code, **insert your CAD bill(s)**.
2. **Confirm the amount on the screen** to complete your purchase.

Please note that there will be a service fee, which varies based on the Bitcoin market.

STEP FIVE: RECEIVING YOUR ETHER

Once you've confirmed your purchase, your ether will be sent to the address you provided*.

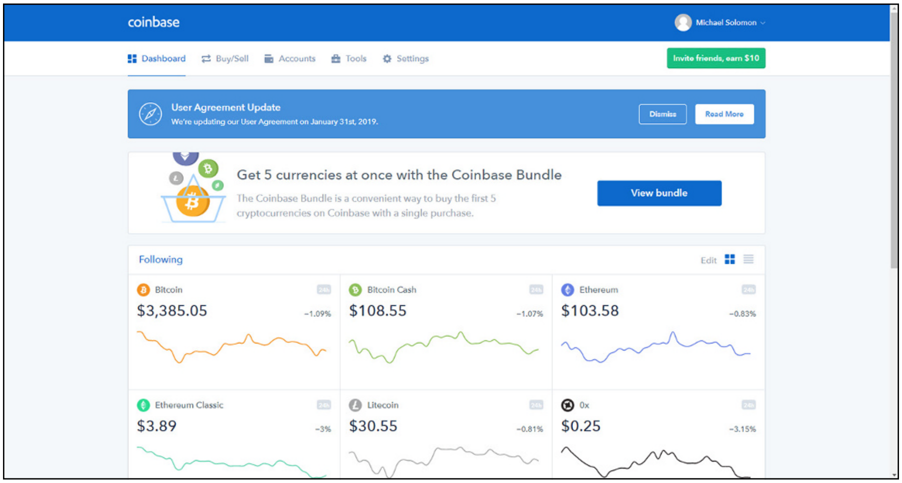
Transactions are usually instantaneous, but may take anywhere between 10–30 minutes, or even 1–2 hours in rare cases**.

Transactions at our ATMs are 100% anonymous.

* If you are going to send funds to another person, we recommended sending it to your own wallet

**FIGURE 1-2:** Purchasing ether with cash.

After you own ether, you can interact with other Ethereum accounts and send them some of your ether in exchange for good or services. Or you can simply hold on to your ether in hopes that is goes up in price. Ether, along with other cryptocurrencies, fluctuates in price continuously. Many investors buy and sell cryptocurrencies as investments, just like trading fiat currencies or commodities. Figure 1-3 shows the main coinbase.com dashboard with popular cryptocurrency prices.



User Agreement Update

We're updating our User Agreement on January 31st, 2019.

Get 5 currencies at once with the Coinbase Bundle

The Coinbase Bundle is a convenient way to buy the first 5 cryptocurrencies on Coinbase with a single purchase.

**FIGURE 1-3:** Current cryptocurrency prices.

At its highest price, ether sold for around $1,400. At the time of this writing, it was down near $100. Whether cryptocurrency is a good investment depends on your appetite for risk and belief in its long-term value.

In addition to buying and trading ether, you can spend it just like any other currency. Of course, you generally have to buy from a vendor that accepts ether. Several service providers make it easy to accept payments with ether, such as Pay with Ether. This company provides the software and the services to make it easy for vendors of any size to accept ether as payment. Visit `www.paywithether.com/` to find out more about this payment option.

TECHNICAL STUFF

There are ways to spend cryptocurrency at vendors that don't directly accept it. Several companies are planning to offer Visa cards that you fund with cryptocurrency. One company, Wirex, allows users to convert their cryptocurrency to USD, GPB, or EUR and use their card at any vendor that accepts Visa.

Cryptocurrency is growing rapidly, but only a small number of vendors accept it. If you really want to pay with ether or other cryptocurrencies, take a look at TenX. This company offers a popular Visa card funded by cryptocurrency. The card isn't available everywhere, but the company expects to increase its availability over time. Navigate to `https://tenx.tech/en` for more information on TenX and their cryptocurrency payment card.

# Getting Started with DAO and ICO

Blockchain technology has given rise to new classes of organizations and opportunities. You'll often hear about *decentralized autonomous organization (DAO)* and *initial coin offering (ICO).* These terms simply describe endeavors that Ethereum makes possible. You'll read a lot about these terms as you learn more about Ethereum, so it makes sense to cover them here.

A *DAO* is an organization that operates only on the rules set forth in its smart contracts. In reality, most DAOs require some human interaction, but the majority of the functionality is automated. For example, assume in just a few years that autonomous vehicles (driverless cars) are more common. A DAO would be like a driverless Uber or Lyft car. The car waits for a passenger, and then drives to the pickup location when someone needs a ride. The autonomous car completes the trip and the passenger pays with cryptocurrency. The car just earned some money. However, the car's maintenance smart contract detects that the brakes need replacing. So the car drives itself to a mechanic and pays for new brakes, using the profits of previous rides. The autonomous vehicle does not need human interaction to carry out its primary business function or to get necessary service. The autonomous vehicle is the same idea as a DAO.

A DAO conducts business and engages in transactions without requiring human interaction. Today's DAOs are relatively simple, but it is expected that they will grow in complexity and eventually replace (or at least compete with) some existing human-based businesses.

Like all businesses, Ethereum-based or Ethereum-related businesses need funding to operate. Many traditional methods for raising funds exist, including soliciting private investors, securing loans, or selling shares in the company. In addition, Ethereum opens new options for funding businesses.

Businesses that use Ethereum often create their own tokens, also called coins, that represents value associated with the business ventures. Businesses sell these tokens to raise funds to launch the business. These *ICOs* essentially exchange one type of currency for a digital item of value. Tokens may represent an expected future value as ownership in a new venture or current value that entitles the holder to some benefit. Either way, tokens are similar in some ways to stock shares. An ICO is a popular method to fund a new blockchain-based business. If you want to learn more about the most popular ICOs, navigate to `www.coindesk.com/ico-tracker` to explore coindesk's ICO Tracker, shown in Figure 1-4.