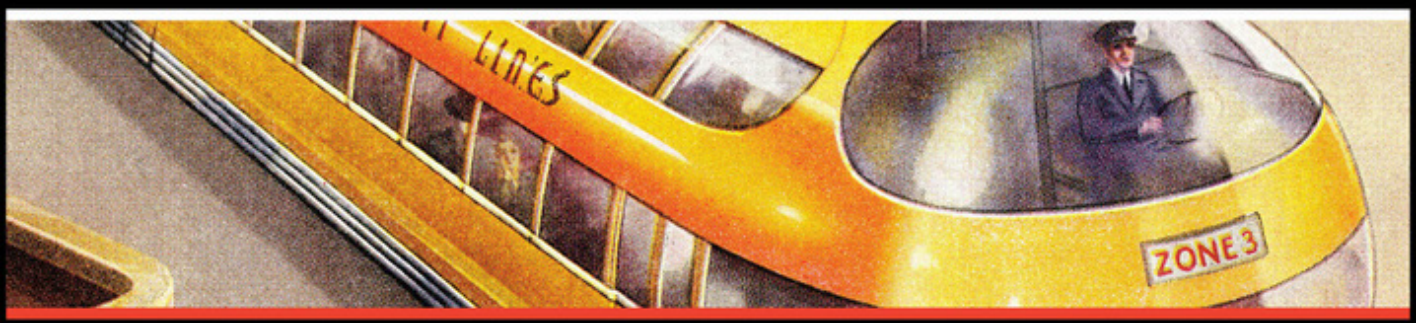
TECH TODAY

BEGINNING SOLIDITY

Learn to Program Smart Contracts with Solidity



ALEXANDROS DOLGOV

WILEY

BEGINNING SOLIDITY

BEGINNING

Solidity

BEGINNING

Solidity

LEARN TO PROGRAM SMART CONTRACTS WITH SOLIDITY

Alexandros Dolgov

WILEY

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada and the United Kingdom.

ISBNs: 9781394290611 (Paperback), 9781394290635 (ePDF), 9781394290628 (ePub)

Library of Congress Control Number: 2025930523

Cover image: © CSA-Printstock/Getty Images

Cover design: Wiley

To my parents for sacrificing a lot for me.

To Satoshi, whoever and wherever you are, for the most potent idea I have ever encountered.

To Hamid Rahimifar for lending me an ear when I most needed it.

To Gary Weimberg for the wonderful conversations and board game rounds.

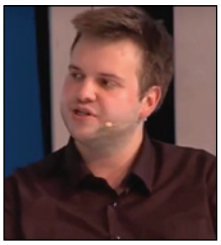
To George Giaglis and the professors at the University of Nicosia for being the first university in the world to create an academic MSc in Blockchain and Digital Currency in 2014 and for teaching me.

To Vitalik, thank you for giving me something to write about.

Weirdly enough, to my life: to the Soviet Union, whose stories from my father about corruption and the economy made me realize the value of Bitcoin.

Finally, to Professor Philipp Sandner. Gone, but not forgotten.

ABOUT THE AUTHOR



Alexandros Dolgov from a young age, questioned why citizens were confined to using national currencies, wondering why they couldn't choose alternatives, whether foreign or self-created. Growing up amid Greece's financial turmoil and influenced by his father's stories of economic collapse in post-Soviet countries, Alexandros developed a deep skepticism of centralized systems and their susceptibility to corruption. His discovery of Bitcoin, DeFi, and DAOs solidified his vision of a fairer, decentralized world, and he saw Solidity as a key tool for building this future.

To deepen his expertise, Alexandros earned an MSc in Blockchain and Digital Currency from the University of Nicosia—the first university in the world to offer this degree—specializing in token economics, decentralized governance, and blockchain technology. He also holds the Chartered Blockchain Analyst Certification. Professionally, Alexandros has contributed to key areas of the blockchain ecosystem, including DeFi, decentralized identity, and token economics. As a guest lecturer at Karel de Grote, he shares his knowledge with future leaders, while his newsletter explores crypto-currencies, Web3, and the fight against censorship and surveillance.

Motivated by a passion for education and advocacy, Alexandros now brings his expertise to this book on Solidity, helping readers unlock the potential of blockchain development to create innovative, decentralized solutions.

Alexandros can be found on LinkedIn at `https://www.linkedin.com/in/alexandros-ntolgkov` where he shares his expertise, on his LinkedIn newsletter where he writes new articles and shares news that have to do with blockchain and technology in general at `https://www.linkedin.com/newsletters/7006379762809978881`, and on X with the handle `@AlexDolgov93`.

ABOUT THE TECHNICAL EDITOR

Wim Van Renterghem is a key lecturer and researcher at Howest, where he plays a pivotal role in developing and teaching courses on blockchain and Web3 technologies. His dedication to continuous professional development is exemplified by his lifelong learning approach, including two microdegrees in blockchain and cybersecurity, and an MSc in Blockchain and Digital Currency from the University of Nicosia, demonstrating his commitment to staying at the forefront of technological advancements.

Wim Van Renterghem is deeply engaged in pioneering research at Howest, focusing on expanding the practical and theoretical boundaries of blockchain technology. His work in various research projects, including BLING and SecuWeb, showcases his innovative approach to integrating blockchain solutions into public services and enhancing web security. These initiatives underscore his ability to tackle complex challenges and drive forward the adoption and refinement of blockchain applications in both governmental and private sectors. Wim's contributions not only advance the technological landscape but also foster a deeper understanding and wider acceptance of blockchain capabilities in real-world scenarios.

ACKNOWLEDGMENTS

I am grateful to the University of Nicosia for its trail-blazing initiative to create an MSc in Blockchain and Digital Currency way back in 2013 when most people did not even know what a cryptocurrency was and most universities in the world laughed at UNIC for launching this degree. Constant year-over-year teaching of the program improved it. When I came along, it made me fall in love with blockchain technology even more than I was.

I am fortunate to have had wonderful teachers and professors at the University of Nicosia such as the one and only Professor George Giaglis, the legendary Andreas Antonopoulos, the UNIC CEO Andreas Polemitis, Lambis Dionysopoulos, Charis Savvides, Apostolos Kourtis, Soulla Louca, Elias Iosif, Marinos Themistocleous, Periklis Thivaios (even though he was the professor I overwhelmingly disagreed with), Christos Makridis, Ifigenia Georgiou, and last but not in any way least Klitos Christodolou. Additionally, I would like to thank Hazal Aripinar, Armantos Katsioloudes, and Andreas Michail.

I am moreover grateful to Wiley for giving me the opportunity to write the book and to the entire editorial team that worked hard to make this book come to life: Tom Dinse, Kenyon Brown, Navin Vijayakumar, Wim Van Renterghem, Kim Wimpsett, and Archana Pragash. I would like to give a warm thanks to Anne Jungers for introducing me to the Technical editor of this book, Wim.

I want to give special thanks to Alexandra Overgaag, CEO of Thrilld Labs, who provided emotional support to me, asked interesting thought-provoking questions, and answered some questions that I had from a reader's perspective during the process of writing this book.

I am lucky to live during a time when Bitcoin was released by Satoshi. Such events—a complete pseudonymous stranger releasing a technology and then disappearing for what seems to be forever without anyone knowing his true identity for sure, and a few years later, a gifted child like Vitalik showing up to come up with the idea of Ethereum—are scenarios that could easily have been in a fictional movie rather than events that happened in real life.

CONTENTS

INTRODUCTION

AT ITS CORE, THIS BOOK IS ABOUT LEARNING Solidity programming, specifically on the Ethereum blockchain, using the Foundry development framework. The goal is to create a resource for seasoned, aspiring, and beginning programmers to make their first steps in mastering the Solidity programming language. Foundry as a framework was selected over Hardhat and Truffle because Foundry is a straightforward framework when compared to those development frameworks. It is written in Rust, and it excels in speed, performance, and being lightweight. Additionally, when using Hardhat and Truffle, you need to be familiar with another programming language, JavaScript, to run tests and deployments. In Foundry, all this is done only through Solidity and cheat codes. The Foundry does not need any additional setup or plugins to work.

Foundry comes with its own native set of tools, such as Anvil, which is Foundry's own local Ethereum development node that helps simulate Ethereum's blockchain environment for testing and debugging. Another tool native to Foundry is Cast, which is a tool that uses the command line to interact with the Ethereum network and allows its command to interact with the different smart contracts that have been deployed on Ethereum's network. Additional interactions using Cast include sending transactions and retrieving blockchain data. The final component in Foundry's arsenal of tools is called the Forge. Forge is Foundry's smart contract development tool, which allows for smart contract compilation, testing, and deployment in Solidity without the need to use JavaScript or any other language.

WHAT THIS BOOK COVERS

When planning and writing this book, my aim was to go beyond simply teaching Solidity in isolation. I wanted to ensure you will gain an understanding of the broader context, including the implications, potential, and history of this technology. To do that, I begin the book by covering topics that, while not directly focused on Solidity, are crucial for building a well-rounded understanding of Ethereum and the underlying technology. Later chapters cover specific tools and techniques you need to know to begin to master Solidity programming.

The first chapter briefly discusses the history of money and how different forms of money have been perceived and used by the cultures that gave birth to each specific type of money. It provides brief overviews from barter and primitive money to today's cryptocurrencies while giving different use case examples of Ethereum's technology.

The second chapter goes briefly through Ethereum's architecture without doing a deep dive into it. It covers the basics of Ethereum's architecture, how it works, and why some decisions about its architecture were taken in the way they were taken. It also explores Ethereum's future developments to help you stay up-to-date with the state of Ethereum and with the planned future developments of the technology.

The third chapter is an introduction and tutorial on installing and using MetaMask, one of the most well-known wallets in the cryptocurrency industry. It covers using Ethereum faucets to get testnet ether—a fake version of Ethereum's cryptocurrency used for testing—and how to use block explorers to view and analyze transactions that happen on the Ethereum blockchain. This chapter also explains how developers can understand the smart contracts they interact with and interpret the information provided by block explorers. It also includes tips on avoiding being scammed by malicious parties and an overview of the different wallets currently in cryptocurrency. This chapter gives developers an understanding of the differences, advantages, and disadvantages of each type of wallet.

The fourth chapter of the book introduces Remix, a native online—and now available offline as well—Ethereum integrated development environment (IDE) that developers and aspiring developers can use to quickly test their smart contracts or take their first steps in Solidity programming. Within the fourth chapter, you will build your first simple smart contract.

In the fifth chapter, while still using and exploring new features the Remix IDE has to offer, you will start building more complex smart contracts, in this case, a zoo management contract that manages the animals of a small zoo as well as the number of visitors to the zoo.

The sixth chapter covers installing a professional IDE called Visual Studio Code (VS Code), which existing programmers know, but aspiring Solidity programmers might not. It also covers installing the Foundry framework to VS Code itself.

The seventh chapter of the book goes over the zoo management contract again, this time in a VS Code and Foundry environment. The main aim is to go through the Foundry framework and showcase its tools, how to use the Foundry tools, how to compile the contract within the framework, and how to deploy the contract both on Anvil's local integrated blockchain that comes with Foundry and on the Ethereum blockchain. It also covers how to manage your private keys when using Foundry and working on smart contracts.

The eighth chapter teaches you how to create a new smart contract. This time, instead of using Remix, you use VS Code and Foundry. The smart contract is a fundraising contract that allows someone to send money to the contract and for the owner of the contract to withdraw the money. The chapter also introduces blockchain oracles, a tool that allows blockchains to communicate data to and receive data from the real world, such as the prices of different currencies and assets.

The ninth chapter explains what an ERC-20 cryptocurrency is, its characteristics, and what ERC itself stands for. It also explains how proposals to change something in Ethereum are created, what they consist of, and, finally, how you can, step-by-step, create an ERC-20 cryptocurrency on your own.

The tenth chapter explains what stablecoins are, the types of stablecoins, how they work, and the mechanisms each uses to keep its price relatively stable. Finally, it explains how to create a stablecoin and a protocol step-by-step that allows the protocol user to borrow that stablecoin.

In the eleventh chapter, you will learn what nonfungible tokens (NFTs) are and how to create one. The chapter also introduces the Interplanetary File System database, a way of storing files worldwide in a decentralized manner to ensure censorship resistance for the NFT collection a developer might create.

The twelfth chapter explains upgradeable smart contracts—ways of upgrading and updating already developed and deployed smart contracts—and teaches you how to implement them.

The thirteenth chapter explains decentralized autonomous organizations (DAOs) in the context of blockchain and cryptocurrencies and teaches you how to create a DAO.

The fourteenth chapter discusses methods of keeping a smart contract secure, explains what a smart contract audit is, and details the process for conducting one.

USING AN AI TOOL

While reading the book, you can benefit from buying a subscription to an AI tool such as ChatGPT, Claude, or other tool to ask questions, request additional information, and enhance your learning. Questions and requests you might ask in AI prompts include but are not limited to asking the AI to break down complex concepts from ERC-20 to ERC-721 to DAOs, upgradable contracts, and others. In addition, you can ask the AI tool to provide analogies for complicated concepts to make it easier to understand something, which can be especially useful for complete beginners to programming, Solidity, and blockchain. Moreover, you can copy-paste into an LLM error messages and the code of a contract from this book and ask it to identify problems and how to fix them. Asking for clarification is crucial for learning and finding solutions.

Every book has limits in terms of scope, and this is no different. In addition, it is impossible to work through all the smart contracts imported and inherited through the different libraries you learn about in this book. A beyond-the-book exercise you can do is to ask the AI to explain snippets of code that are not examined in the book to enhance your understanding and deepen your knowledge and skill of not only writing code but also reading it and comprehending what a piece of code likely does.

Another way of using an AI companion is to ask it to create additional questions based on the book's material or something closely related to it and answer the questions to ensure a deep understanding of the material. Finally, the best way of using AI is not in the way that I am suggesting you use it—although this has its uses, too—but in how you come up with using it to enhance your learning. Think and be creative with how you use AI; try your ideas with prompts and see how effective they are.

To conclude this introduction, it is crucial to point out another limitation of books: they are usually slower to come up with new editions and updates than the industry and subject matter they delve into, making them a bit out-of-date after a couple of years. To minimize this as much as possible, it could be useful to go through the documentation of the tools and frameworks used in this book such as Foundry and Solidity itself to ensure that you stay up-to-date with any updates that might render some parts of the codebase in this book obsolete.

- **Foundry documentation link**: `https://book.getfoundry.sh`
- **Solidity documentation link**: `https://docs.soliditylang.org/en/latest`

Finally, on this book's website you'll find all the code available for download so you can work and experiment with the live code. The files are at:

`https://www.wiley.com/en-us/Beginning+Solidity%3A+Learn+to+Program+Smart+Contracts+with+Solidity-p-9781394290611`

1

What Is Money and a Brief History of It?

This chapter explores the fundamental concept of money and its evolution through human history, providing essential context for understanding blockchain-based financial systems and smart contract development. By examining how money functions as both a technology and a language for value exchange, you'll gain insight into why decentralized systems like Bitcoin and Ethereum represent the next evolution in monetary and contractual relationships.

The chapter addresses key questions: What makes something valuable enough to serve as money? How have societies historically solved the coordination problems of value exchange? How do cryptocurrencies and smart contracts build upon these historical patterns while introducing new capabilities?

Through detailed analysis of primitive money systems, modern banking, and emerging crypto networks, you'll learn how money's core properties—medium of exchange, store of value, and unit of account—manifest across different technological implementations. The chapter connects these concepts directly to smart contract development by examining real-world applications ranging from tokenized real estate to decentralized autonomous organizations (DAOs).

By working through historical examples and modern case studies, you'll develop a crucial foundation for understanding the problems that smart contracts aim to solve and the new possibilities they enable. These insights are essential for designing effective decentralized applications and financial protocols that align with fundamental economic principles while leveraging blockchain's unique capabilities.

WHAT IS MONEY?

This, first and foremost, is a Solidity book. However, money is integral to cryptocurrencies, to blockchains such as Ethereum, to Solidity, and to smart contracts. Thus, a short chapter on

money in this book is essential for you to fully and deeply understand and appreciate the bigger picture of what it is that people are, in essence, building when they use Solidity to create smart contracts. In this chapter, we'll cover the history of money for a deeper understanding of Ethereum, Solidity, smart contracts, and cryptocurrencies in general.

Money has many meanings and definitions. Money has been called a medium of exchange. This commodity represents the time and energy given to an employer to build or do something. In exchange, we get money. Another way of defining money is as energy—spent in service of an employer or on creating a service or a product—that takes physical form as coins and paper notes that can then be used in exchange for someone else's energy, who was spending their energy building a car, organizing a vacation, working at a hotel we visit, preparing the meal we would eat, and so forth. These are just a few definitions of money throughout history.

Since this is a technology book, we will look at money not as a commodity but as a language and technology, seeing the current global economy—and those before it—as a protocol built on this language and technology. It is estimated that there are around 7,000 languages in the world that people use to communicate with each other every day. There are around 700 programming languages in the world. When it comes to computers, they fundamentally understand only binary sequences of 0s and 1s, with each separate sequence of 0s and 1s meaning something entirely different from the others. Money, on the other hand, no matter the currency, is a language humans use to communicate how much they value a product or a service. On the most basic level, money is saying, "Oh, you have a great kitchen knife for sale there. I value it at 50 amounts of money. Will you sell it to me?" (Replace "amounts of money" with dollars or euro or the currency of your country.) If the person selling the kitchen knife values the labor and time that went into creating the kitchen knife to 50 amounts of money, they will say "yes," and the kitchen knife will be sold. In this example, money acts as a language between two parties, conveying the value they assign to a specific item or service—in this case, a kitchen knife. Before finalizing the transaction, the parties use this shared "language" to communicate and agree upon the knife's value. Once agreed, the transaction proceeds: the buyer transfers the agreed-upon value to the seller, and the seller transfers ownership of the knife to the buyer.

Money as a form of technology—and the economic systems built upon it as protocols derived from this "language"—is a complex and expansive subject. It provides material spanning bachelor's, master's, and PhD programs, as well as countless books and even entire series of books, depending on the depth of exploration. In this chapter, however, we will take a condensed approach, focusing on the major milestones in the evolution of money as a technology. This will involve simplifying and omitting many details to provide a high-level overview.

Money as a Technology

Money as a technology is a contract in and of itself in two ways:

- A contract is an agreement among a community of people to use a specific type of money as a medium of exchange to transact with one another for goods and services. This agreement establishes their willingness to treat this form of money as the community's transactional tool.
- A contract between two parties that are not part of the same community needs a standard transactional tool to exchange goods and services. This ensures the transactional tool will be

valuable to the party selling the goods and services. The following are two examples of that, one historical and one contemporary:

- **Historical:** Members of two different tribes engaging in commerce with one another—with each tribe using its type of money that may not be valuable to the other—agreeing to use another intermediary for commerce, such as coins, belts, animal skins, or anything else that may be of value to both tribes.
- **Contemporary:** Modern countries have agreed to use the US dollar (USD) as a medium of exchange for international business transactions. Before USD became the world's reserve currency, countries agreed to use the British Pound Sterling as a medium of exchange for international transactions. Before the pound, it was the Dutch guilder; before that, the Spanish Real; and before even that, the Italian Florin.

When it comes to using money as technology, money has three properties, each having a separate set of subproperties to fulfill for that property to be considered viable as use for money. Those three properties and their subproperties are discussed next.

Medium of Exchange

A *medium of exchange* is an intermediary technological tool between parties that can facilitate value communication in exchange for products and services. In turn, the party that exchanged their service or product for the medium of exchange can use that medium of exchange to buy the products and services that they need from someone else in an endless, circular way until the specific medium of exchange used, be it coins, paper money, or digital money, is destroyed by being taken out of circulation or is damaged irreparably to the point that no party would accept it as a medium of exchange. In the modern era, the currency issued by a country's central bank, also known as the national currency of a country, is the primary medium of exchange for participating in an economy and exchanging goods and services.

For a medium of exchange to be *reliable*, it must have certain subproperties:

- **Durability:** This is the extent to which a currency can be used without significantly being damaged or torn.
- **Transportability:** How easily can a currency be transferred from one place to another in terms of speed and weight?
- **Divisibility:** Can the money be divided into smaller chunks? Can a 100 USD bill be divided into smaller chunks of 50, 20, 10, 5, 2, and 1 USD and then into cents?
- **Fungibility:** Is money fungible? Following the earlier dollar example, is a 100 USD bill equal to another 100 USD bill?
- **Noncounterfeit ability:** How easily can the currency be counterfeited?
- **Scarcity:** Is the medium of exchange scarce enough in supply to ensure that there is value in it? In other words, is it rare enough to come by so that people want it? When a medium of exchange is available in abundant supply, it loses value. With a "reasonable" decrease in scarcity and consequent loss of value, people raise the price of their goods. When the surplus

increases significantly, the money loses its value to the extent that it becomes useless, and nobody wants to use it anymore.

- **Acceptability:** The more acceptability a medium of exchange has, the more effective it becomes. Its acceptability can vary from one person to another or one store or merchant to another. The acceptability of a medium of exchange can be as narrow as only two parties agreeing to use something as a medium of exchange and transact with it. This would be extremely weak acceptability, but it is still nonetheless something that can be considered as acceptability of a medium of exchange. That said, when a medium of exchange is desired, its acceptability becomes much more potent and much more widely acceptable than just two parties, making it a more effective medium.

Store of Value

Another significant property of money is as a store of value. A *store of value* is a tool or property of money that allows for money's value to be saved and retrieved in the future with a degree of predictability about the value remaining, or at the very least, the same in the future. It is worth noting that a store of value is not a property of only money but also of assets, as there are different stores of value, such as gold, silver, diamonds, reserve currencies, government bonds, stocks, real estate, and other assets.

The property of being a good store of value is derived from the following:

- **Stability:** Current predictions of a stable or predictable demand for an asset, be it money or any previously mentioned assets.
- **Predictability:** An asset's stable or shrinking future supply. As all assets and currencies have a certain degree of unpredictability in value, today there is no perfect store of value.
- **Cultural and societal values:** The perceived cultural value of certain currencies or assets that may inflate their price beyond the apparent use in industrial or other uses.
- **Liquidity:** The ease with which a store of value can be converted into a medium of exchange without loss—or significant loss—in value. High liquidity means that an asset can be quickly sold or at least converted for cash (or its equivalents) when required, which enhances its utility as a store of value. For example, suppose you buy real estate and, after a few years or months, want to sell it. It could take months or possibly years to sell the real estate and convert it into money or cash. On the other hand, stocks can be sold much faster, increasing their liquidity and conversion into cash.
- **Portability and transferability:** Like the transportability of a medium of exchange mentioned earlier, how easy it is for someone to transfer the assets to someone else, from one party to another, or from one geographical location to another, without any compromise in value and reduced costs. This is, of course, much easier to do when a proof of ownership document—whether digital or physical—is available. In contrast, transporting 100 bars of gold poses numerous challenges, such as moving them from one US state to another or, even worse, from one continent to another. Such a scenario increases costs and exposes the stored value to risks including damage, theft, or other unforeseen issues.

Unit of Account

The final significant property of money is that of a *unit of account*. For a type of money to be considered a unit of account, prices must be communicated in terms of money and currency rather than other goods. For example, "I paid 10.000 USD for a trip to Europe" instead of "I paid four cows, or 66 shares of Alphabet (Google) stock for a trip to Europe." In money's language, prices in a unit of account properly communicate the value of goods and services, assets, liabilities, and other economic activity. In each country, the main currency tends to be the unit of account; certain exceptions apply, such as the European Union, a union of countries bound by a single currency, the euro.

For money to be considered a practical unit of account and used to communicate value for goods and services to others, it must be stable in price and not volatile. In the short term, national currencies such as the US dollar and the euro are much more stable than cryptocurrencies.

However, in the long term, the current structure of the global economy integrates inflation and the practice of printing money mainly due to money's endless supply as a natural aspect of annual economic operations. Inflation and the practice of money printing are seen as unavoidable in the current economy and desirable in activating the economy and stimulating spending by aiming for a 2 percent annual inflation. However, actual rates frequently exceed this target. Increasing the money supply through printing money and thus creating inflation impacts the currency's scarcity, decreasing its value over time.

The preceding discussions of money's properties explain why no form of money today can be considered perfect: none fulfills all the previously mentioned properties. When some of these properties are taken to the extreme, they tend to negate some of the other properties, which suggests that currently the ideal economic state is a balance between these properties while maintaining periodic adjustments toward higher or lower inflation based on the condition of the economy at a given time.

THE HISTORY AND EVOLUTION OF MONEY

Now that you have an understanding of what money is, let's pivot to gaining an understanding of its history and evolution.

Money is a fundamental part of human civilization, evolving alongside societies to meet the needs of trade, value storage, and wealth exchange. This chapter explores the history and evolution of money, from its earliest form—barter—to the development of primitive money, highlighting how humans transitioned from simple exchanges to complex economic systems.

Before currency was invented, people relied on barter within small, self-sufficient communities. Over time, the limitations of barter, such as the "double coincidence of wants," led to the creation of primitive forms of money, including shells, wampum belts, cattle, and even salt. These items were shaped by their cultural and environmental contexts, serving as both mediums of exchange and symbols of social value.

Through this exploration, we see that money is not just a tangible object but also a psychological and social construct. Its evolution reflects humanity's drive to create tools and systems that enable peaceful trade and cooperation, ultimately shaping the economic frameworks we rely on today.

Barter

Before any currency or form of money was invented, people lived in tiny, agrarian, self-sufficient communities where everyone practically knew each other and engaged in the communication of value, shaping of contracts, and trade of services and goods they found valuable through an activity called *barter*, which was simply trading goods and services directly and without having an intermediary tool representing money. Instead of paying to buy a cow with cash, people would trade six chickens for one cow or a few eggs from the chickens for milk from the cow. Since in many of those early communities everyone knew each other, there was no need for written contracts, and in any case contracts were mainly oral and based on honor until writing was invented. After that, people would transcribe their contracts on clay tablets. If someone from the community knew that Person X did not fulfill his part of the agreement with Person Y, their reputation would be at stake. People would refuse to enter into commerce contracts with them out of fear of them not reciprocating their part of the deal.

Barter can be an acceptable approach when an economy is tiny and based mainly on honor due to the people knowing one another. In those circumstances people could exchange the items they needed without the need for money as a technology serving as an intermediary of exchange. Such small economies were possible when humankind lived in caves or small communities. Another possibility could be when humans would start a new settlement or town. At the very beginning of the town's building, there would not be too many people living there, and hence, the villagers or townspeople could, in theory, implement a barter system instead of needing a currency or money to engage in trade of their services and products and satisfying their needs. Another historical example is when people are detained as prisoners for any reason. During World War II, Jewish people—and possibly other prisoners—used tobacco and cigarettes as currency for favors, products, services, bribes, and other economic activities, both ethical and unethical.

Barter as an economic system faces a major limitation known as the "double coincidence of wants." This occurs when two parties must each have something the other desires to make an exchange. For example, if one person trades 6 liters of milk for 12 eggs, both must agree on the value of this trade. However, problems arise if either party needs something else afterward. Suppose the person who now has milk and eggs wants to buy three fish for dinner. They must find a fish merchant willing to accept milk or eggs as payment. If the merchant agrees, they might, in turn, need a haircut. To get one, the merchant must find a barber willing to accept fish, milk, or eggs as payment. This chain of dependencies becomes increasingly impractical. Additionally, while goods like milk, eggs, or fish can be traded, intangible services like a haircut cannot be exchanged further as tradeable assets. This limitation reduces the efficiency of barter compared to monetary systems.

Moreover, there is no guarantee that the barber, fish merchant, or any other two people will want fish, eggs, or milk in exchange for the goods or services they provide. The more people entering the equation—or the more variables there are in a function—the more difficult it is to find someone who wants precisely the goods or services you specialized in to trade his goods and services with you. Furthermore, many products, including eggs, milk, and fish, are not suitable replacements for money or currency used as a medium of exchange or even a store of value since—among other drawbacks—they go bad quickly. They, hence, are not durable or feasible to transfer rapidly in large quantities and through considerable distances.