uphold

My first blockchain in Python

... that does not require you to be an EDP shareholder to run

Agenda

1. Audience check
2. What is Uphold?
3. Who am I?
4. Brief intro to Python
5. Brief intro to Blockchain
6. Coding a blockchain
7. Recap
8. Further reading

How familiar are you with...

How familiar are you with...

🐍 ... **Python**?

How familiar are you with...

🐍 ... **Python**?

₿ ... **blockchain** technologies? (and Bitcoin, Smart Contracts, NFTs, Ponzi Schemes , ...)

How familiar are you with...

🐍 ... **Python**?

... **blockchain** technologies? (and Bitcoin, Smart Contracts, NFTs, Ponzi Schemes, ...)

... **Uphold**?

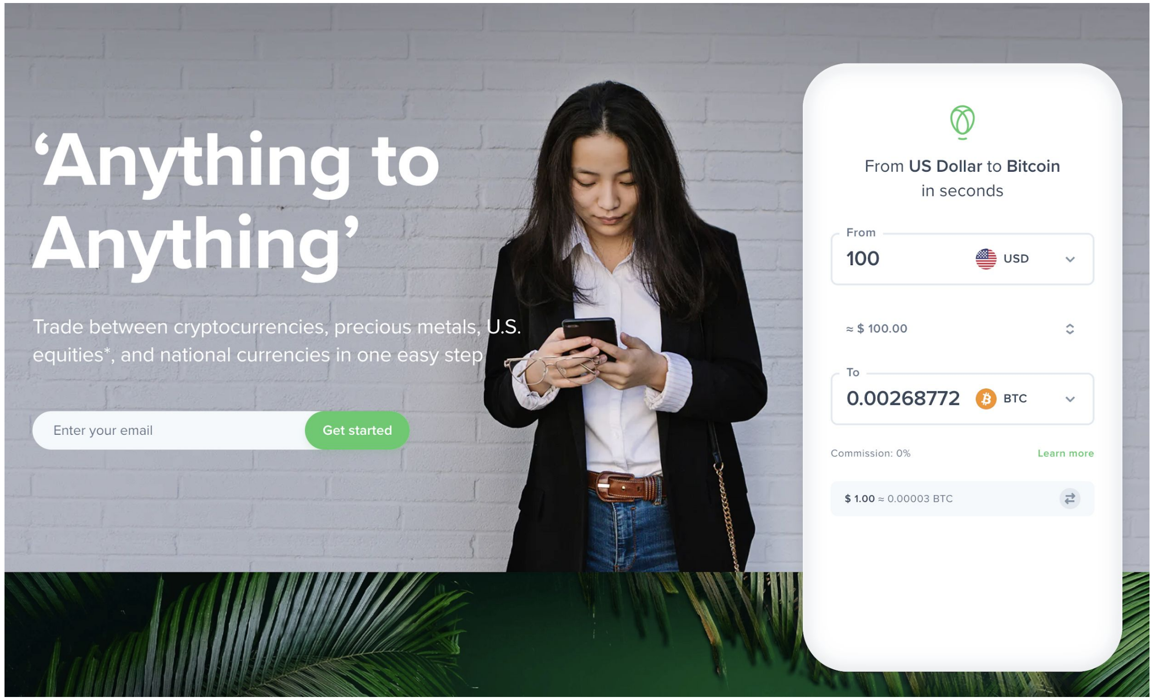
What is Uphold?

Our vision

To provide trusted transparent access to digital money and financial services to people everywhere.

Since our inception, we have fought to provide a fairer, easier and more affordable system for both consumers and businesses. We favor speed, simplicity and ease of use over complexity. We put security and transparency first.

Uphold platform


‘Anything to Anything’

Trade between cryptocurrencies, precious metals, U.S. equities*, and national currencies in one easy step

From **US Dollar** to **Bitcoin** in seconds

From: 100 USD

≈ $ 100.00

To: 0.00268772 BTC

Commission: 0%

$ 1.00 ≈ 0.00003 BTC

Value proposition

Easy to use

We make financial transactions simple to understand and to make.

Access to new financial systems

We provide true transactional connectivity to many cryptocurrency networks.

Safer money

Fully reserved, we protect 100% of our customers' assets in our vaults.

Transparent

Uniquely, we publish all transactions and holding published in real time on the public blockchain.

Our values

Secure

We are guardians of people's money. Above all, it is our duty to protect our members.

Transparent

Trust is an essential and hard won privilege. We are honest and transparent in everything we do.

Innovative

We are constantly curious. We listen to our members and the world around us. We think rigorously to design elegant solutions. And we work intensely to develop and deploy services that bring real value to our customers.

Respectful

We treat our members, partners and team as individuals. We strive to understand and support the many different types of people who touch our products, services and organization around the world.

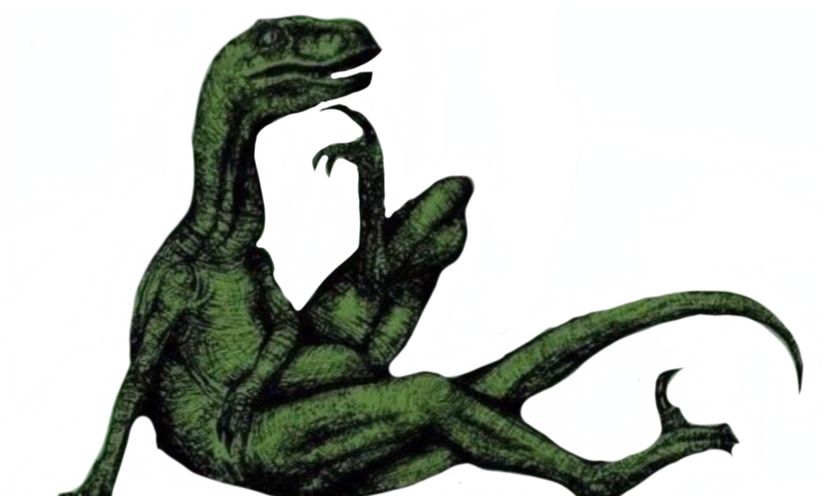
Open

We want our services to reach as many people as possible both directly to consumers and also indirectly via our API and partnership.

And who am I?

And who am I?

A meatbag searching for meaning in a meaningless world.



… ok, but other than that?

- Currently VP of Data Science and Analytics
- Working at Uphold for 3+ years
- Studied Informatics Engineering at FEUP
- 9 years of regular interaction with Python, mostly in the Data Science realm
- Passionate about Philosophy, Economics, Statistics, Machine Learning Parkour, MTG, …

diogojapinto@gmail.com
@diogojapinto

… ok, but other than that?

- Currently VP of Data Scien
- Working at Uphol
- Studied I
- ython, mostly in the Data Science
- hilosophy, Economics, Statistics, Machine Learning
- TG, …

I DO NOT PROVIDE FINANCIAL ADVICE!!!

diogojapinto@gmail.com
@diogojapinto

What is Python?

More than a language, it's an ecosystem

Created by **Guido van Rossum** and released in 1991

Simple and efficient **syntax** – designed for readability

Interpreted – great for **prototyping** solutions

Huge and versatile **ecosystem of libraries**:

- Data Science
  - Complex mathematics
  - Working with files and databases
- Web development
- Anything you can dream of

W3Schools Tutorial

What is a Blockchain?

A clever way to achieve trust without a central authority

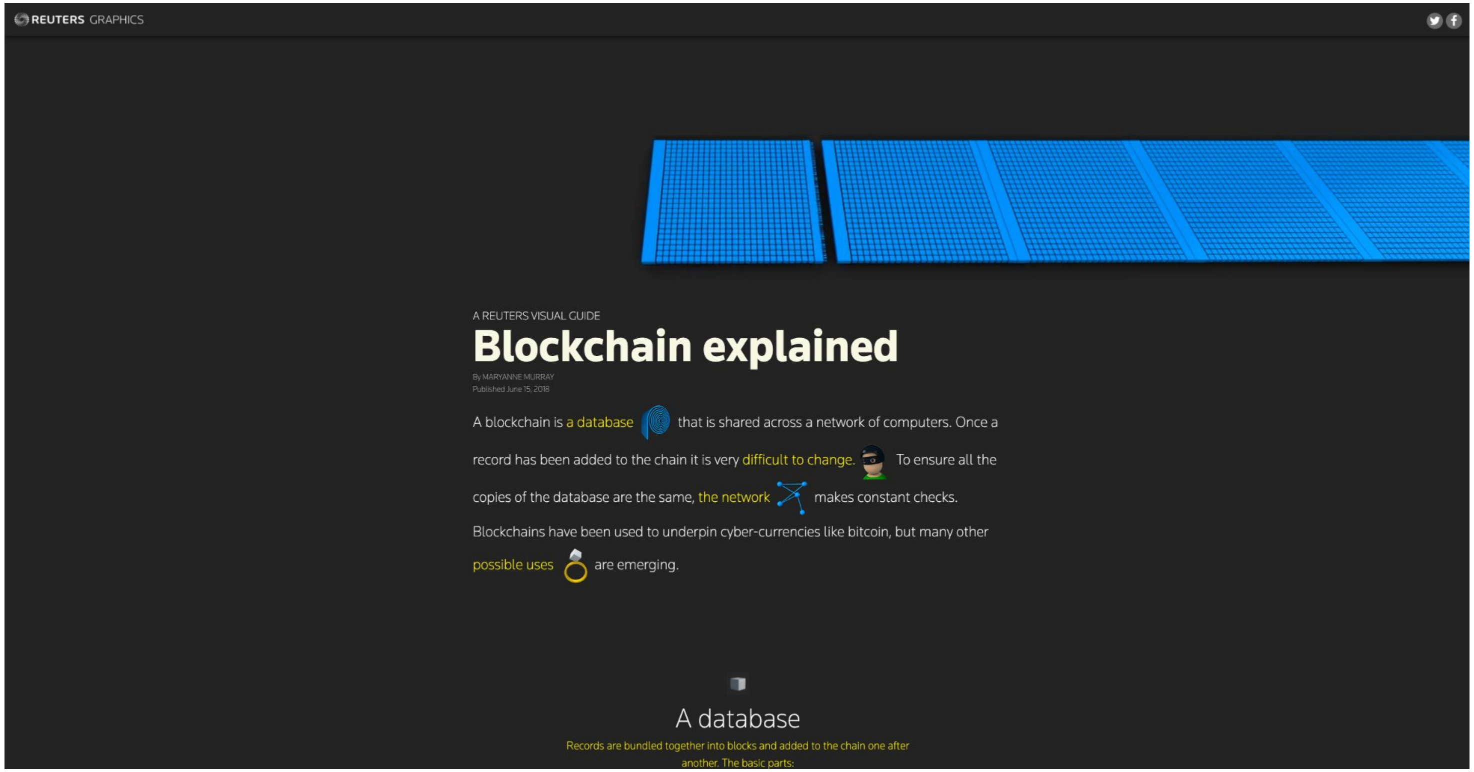
Distributed software, each machine follows a set of rules with no hierarchy

Each block of information is cryptographically secured and connected to each other – **Append-only "distributed" database**

Cryptocurrencies are one of the most widely known use-cases, but it is not the only one:

- Supply-chain
- Insurance
- Lending and borrowing
- Digital identity - Voting
- Proof of ownership (NFTs)
- …

A visual overview



A REUTERS VISUAL GUIDE

Blockchain explained

By MARYANNE MURRAY
Published June 15, 2018

A blockchain is a database that is shared across a network of computers. Once a record has been added to the chain it is very difficult to change. To ensure all the copies of the database are the same, the network makes constant checks. Blockchains have been used to underpin cyber-currencies like bitcoin, but many other possible uses are emerging.

A database

Records are bundled together into blocks and added to the chain one after another. The basic parts:

Let's Code?

Based on [this]() and [this]() articles

1.1. Install Miniconda

```
> wget
https://repo.anaconda.com/miniconda/Miniconda3-latest-Li
nux-x86_64.sh
> chmod +x Miniconda3-latest-Linux-x86_64.sh
> ./Miniconda3-latest-Linux-x86_64.sh
```

1.2. Create environment

```
> conda create --name my-first-blockchain python=3.9
> conda activate my-first-blockchain
> conda install -c conda-forge jupyterlab
```

1.3. Launch Jupyter Lab

```
> jupyter lab
```

2.1. Import required libraries

```python
import hashlib
import json
from time import time
```

2.2. Create "The Blockchain" class

```
class Blockchain(object):
    def __init__(self):
        self.chain = []
        self.pending_transactions = []
        self.difficulty = 5

        self.new_block(previous_hash="The Times 
03/Jan/2009 Chancellor on brink of second bailout for 
banks.", proof=100)
```

2.3. New block creation

```
     @property
     def last_block(self):

         return self.chain[-1]

    def new_block(self, proof, previous_hash=None):
         block = {
             'index': len(self.chain) + 1,
             'timestamp': time(),
             'transactions': self.pending_transactions,
             'proof': proof,
             'previous_hash': previous_hash or self.hash(self.chain[-1]),
         }
         self.pending_transactions = []
         self.chain.append(block)

         return block
```

2.4. New hash function

```
   def hash(self, block):
        string_object = json.dumps(block,
sort_keys=True)
        block_string = string_object.encode()

        raw_hash = hashlib.sha256(block_string)
        hex_hash = raw_hash.hexdigest()

        return hex_hash
```

2.5. New transaction function

```
def new_transaction(self, sender, recipient, amount):
     transaction = {
         'sender': sender,
         'recipient': recipient,
         'amount': amount
     }
     self.pending_transactions.append(transaction)
     return self.last_block['index'] + 1
```

2.6. Testing

```
blockchain = Blockchain()
t1 = blockchain.new_transaction("Satoshi", "Mike", '5 BTC')
t2 = blockchain.new_transaction("Mike", "Satoshi", '1 BTC')
t3 = blockchain.new_transaction("Satoshi", "Hal Finney", '5 BTC')
blockchain.new_block(12345)

t4 = blockchain.new_transaction("Mike", "Alice", '1 BTC')
t5 = blockchain.new_transaction("Alice", "Bob", '0.5 BTC')
t6 = blockchain.new_transaction("Bob", "Mike", '0.5 BTC')
blockchain.new_block(6789)

print("Blockchain:")
print(json.dumps(blockchain.chain, indent=4, sort_keys=True))
```

Issues? 😱

Issues? 😱

No conditions for block creation

No validation of the chain

No validation of transactions submitted for the chain

Still centralized

Issues? 😱

No conditions for block creation

No validation of the chain

No validation of transactions submitted for the chain

Still centralized

3.1. Proof of work function

```
def proof_of_work(self, previous_proof):
     new_proof = 1
     check_proof = False

     while check_proof is False:
         proof_string = str(new_proof**2 - previous_proof**2).encode()
         raw_hash = hashlib.sha256(proof_string)
         hex_hash = raw_hash.hexdigest()

         if hex_hash[:self.difficulty] == '0' * self.difficulty:
             check_proof = True
         else:
             new_proof += 1

     return new_proof
```

3.2. Chain validation function

```
def is_chain_valid(self, chain):
     previous_block = chain[0]
     block_index = 1

     while block_index < len(chain):
         block = chain[block_index]
         if block['previous_hash'] != self.hash(previous_block):
             return False
         previous_proof = previous_block['proof']
         proof = block['proof']
         proof_string = str(proof**2 - previous_proof**2).encode()
         raw_hash = hashlib.sha256(proof_string)
         hex_hash = raw_hash.hexdigest()
         if hex_hash[:self.difficulty] != '0' * self.difficulty:
             return False
         previous_block = block
         block_index += 1
     return True
```

3.3. Mine block function

```python
def mine_block(self):
        previous_block = self.last_block
        previous_proof = previous_block['proof']
        proof = blockchain.proof_of_work(previous_proof)
        previous_hash = blockchain.hash(previous_block)
        block = blockchain.new_block(proof, previous_hash)

        response = {
            'message': 'A block is MINED',
            'index': block['index'],
            'timestamp': block['timestamp'],
            'proof': block['proof'],
            'previous_hash': block['previous_hash']
        }

        return response
```

3.4. Testing

```python
blockchain = Blockchain()
blockchain.is_chain_valid(blockchain.chain)
t1 = blockchain.new_transaction("Satoshi", "Mike", '5 BTC')
t2 = blockchain.new_transaction("Mike", "Satoshi", '1 BTC')
t3 = blockchain.new_transaction("Satoshi", "Hal Finney", '5 BTC')
blockchain.mine_block()
blockchain.is_chain_valid(blockchain.chain)

t4 = blockchain.new_transaction("Mike", "Alice", '1 BTC')
t5 = blockchain.new_transaction("Alice", "Bob", '0.5 BTC')
t6 = blockchain.new_transaction("Bob", "Mike", '0.5 BTC')
blockchain.mine_block()
blockchain.is_chain_valid(blockchain.chain)

print("Blockchain:")
print(json.dumps(blockchain.chain, indent=4, sort_keys=True))
```

Homework time!

Intermediate level:

- Validate that transactions are valid (i.e. non-negative balances)
  - Auxiliary function called in `new_transaction`
- Turn the Blockchain into a web server (e.g. check Flask)
  - Endpoint to add transactions
  - Asynchronously attempt to mine new blocks

Advanced level:

- Make it distributed – launch multiple local threads that communicate between themselves

Food for thought?

What happens if we change the difficulty?

Is the blockchain now secure? How could we hack it?

How can we modify the kind of data we're storing?

Further Reading

[Paper] Bitcoin: A Peer-to-Peer Electronic Cash System

[Video] But how does bitcoin actually work? - 3Blue1Brown

[Video] How does a blockchain work - Simply Explained

[Video] Whiteboard Crypto

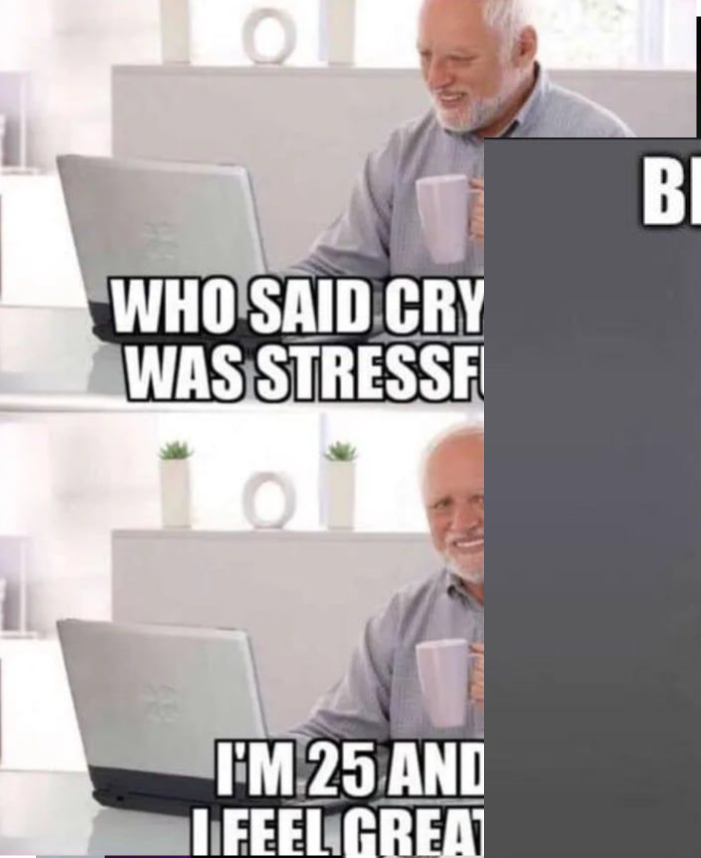
[Video] What are NFTs? | The Economist

[Video] Line Goes Up – The Problem With NFTs

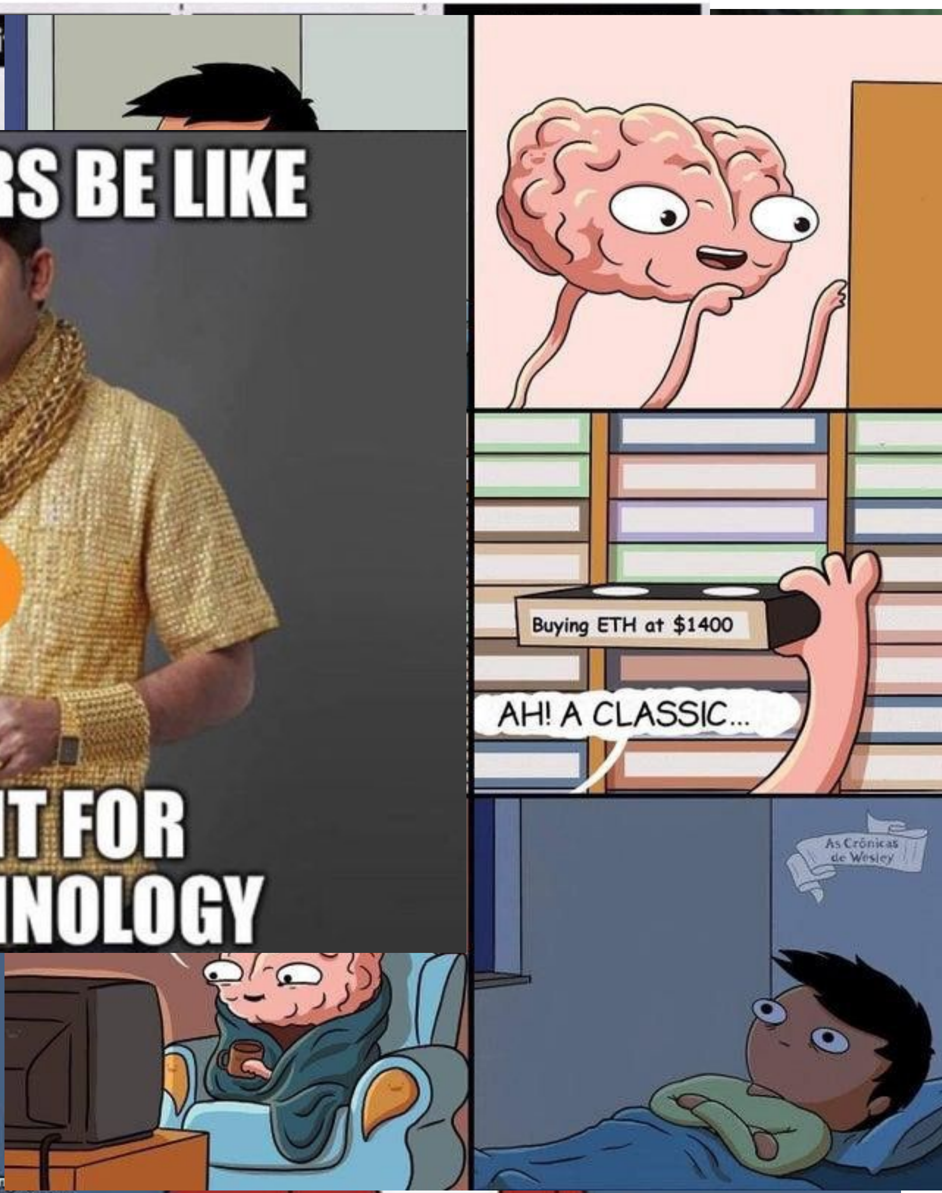
[Podcast] Vitalik, Ethereum Part 1 and Part 2

[Article] Blockchain beyond the hype: What is the strategic business value?

Memes Time!


WHO SAID CRY
WAS STRESSF
I'M 25 AND
I FEEL GREAT

BITCOINERS BE LIKE
I'M IN IT FOR
THE TECHNOLOGY


Buying ETH at $1400
AH! A CLASSIC...
As Crônicas de Wesley

What Politic
LEFT BEFORE HDD SHUTS
DOWN FOREVER
MEMEFUL.COM

2022

github.com/uphold/my-first-python-blockchain

I hope you enjoyed and learnt something!