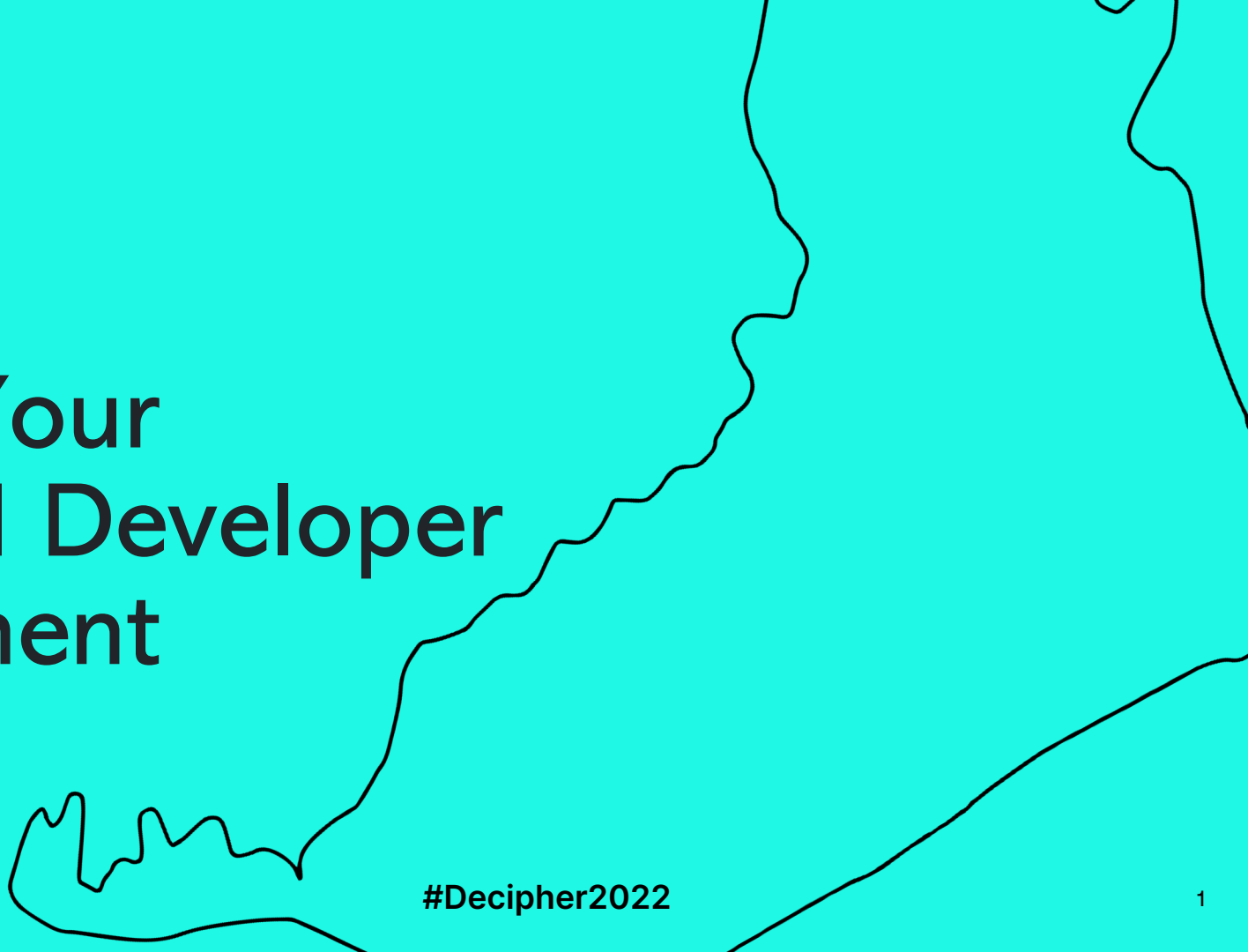
LFG AlgoDevs 🚀🚀

Launch Your Algorand Developer Environment



#Decipher2022

how it started…

Ryan R. Fox

Developer Advocate

@Algorand

…how it's going

@ryanRfox

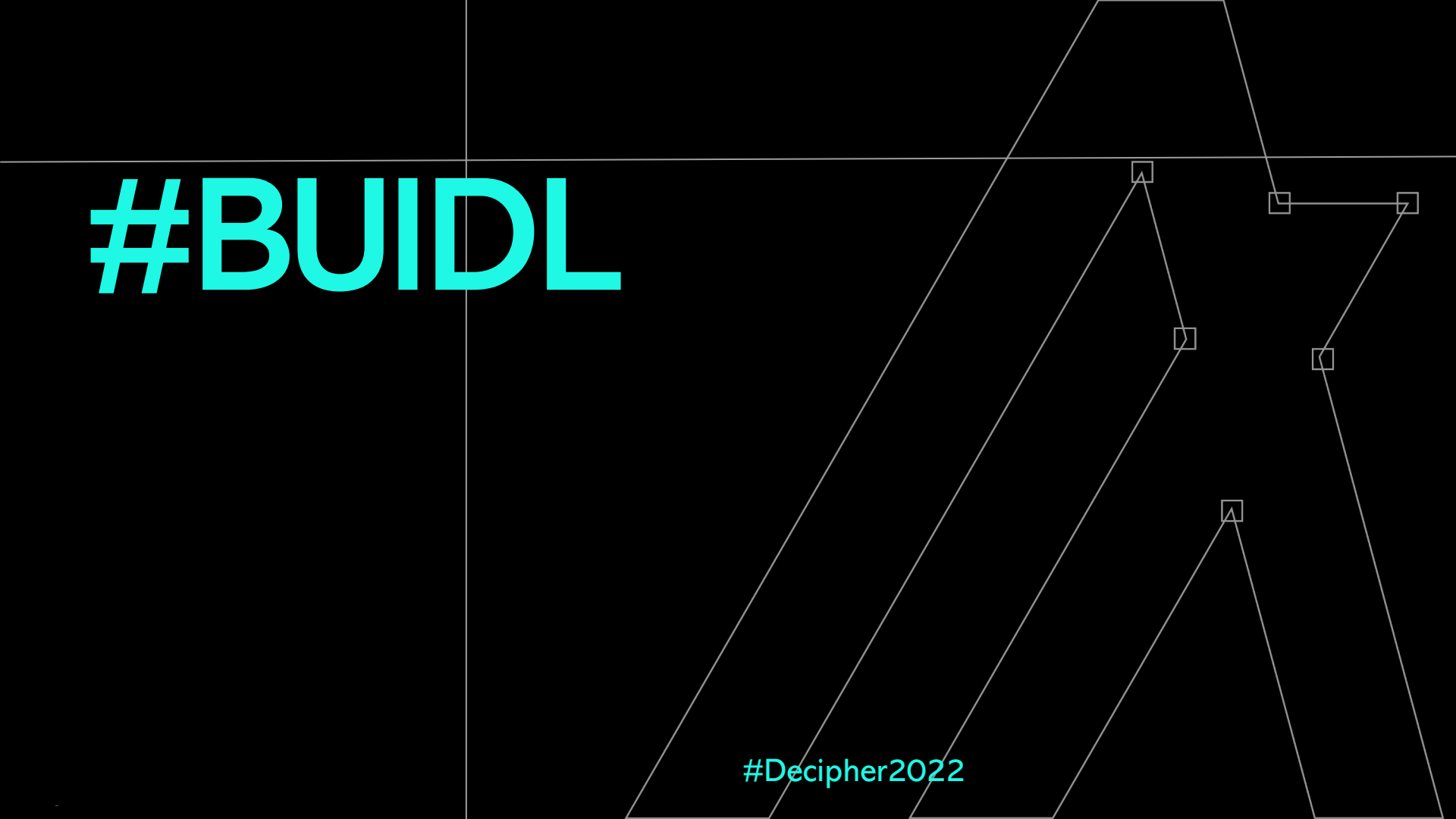
Ryan R. Fox

Developer Advocate

@Algorand

Let’s Connect

#BUIDL



#Decipher2022

#BUIDL

Smart Contracts

#Decipher2022

#BUIDL

dApps

#Decipher2022

#BUIDL

DeFi

#Decipher2022

#BUIDL

Games

#Decipher2022

#BUIDL

Smart NFTs

#Decipher2022

#BUIDL

Randomness

#Decipher2022

#BUIDL

Oracles

#Decipher2022

#BUIDL

Bridges

#Decipher2022

#BUIDL

Community

#Decipher2022

#BUIDL

Web3 Future

#Decipher2022

#BUIDL

Algorand

#Decipher2022

#AlgoFam

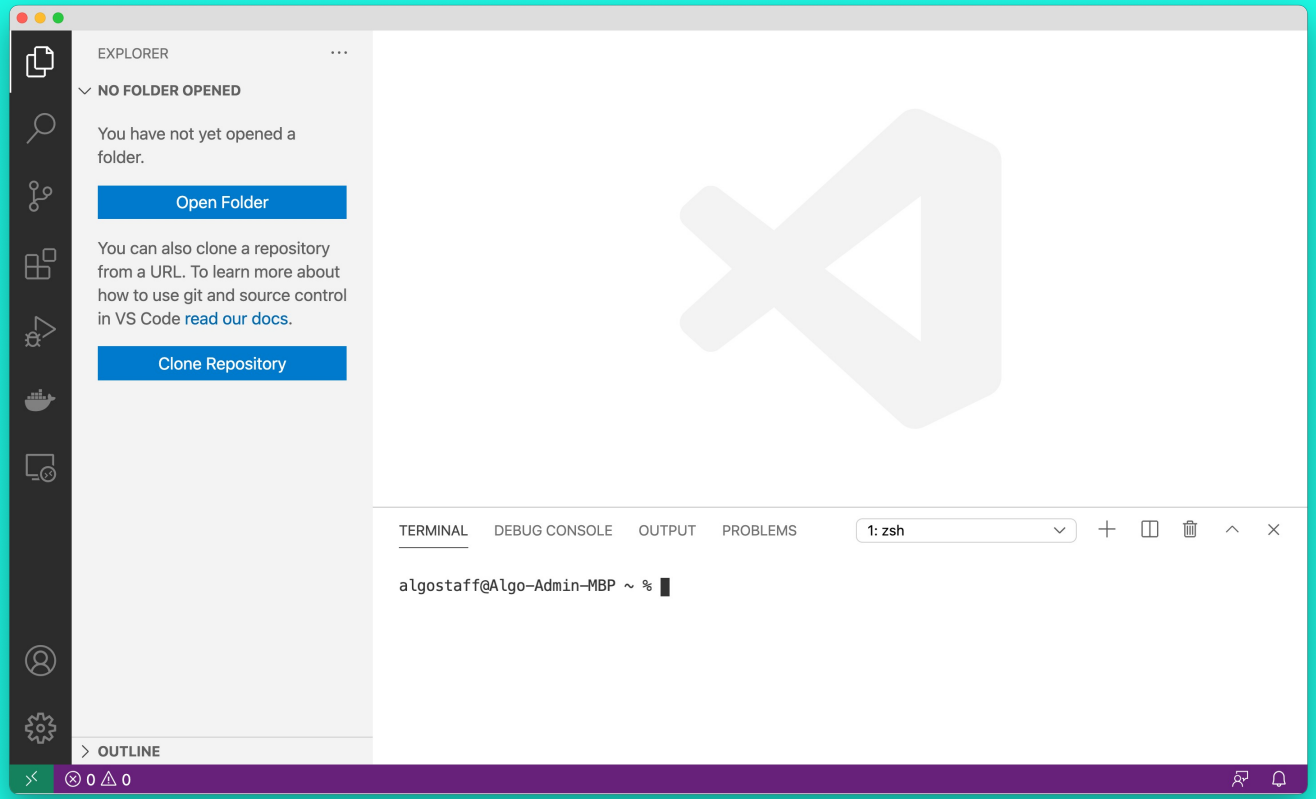
#Decipher2022

@AlgoDevs

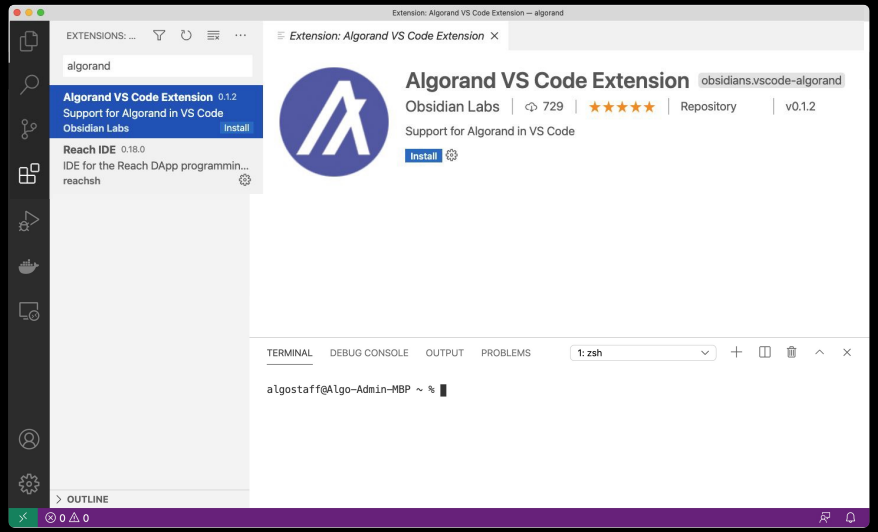
Algorand

#Decipher2022

Developer Environment Setup



Recommendation: VSCode

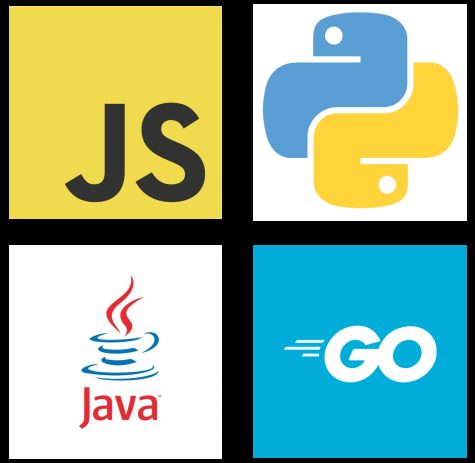


https://code.visualstudio.com/Download

Developer Environment Components

Infrastructure

SDK



Language + Framework

Beaker

$ whoami

pc

mac

macOS prerequisites

Docker Desktop	>= 4.8.0
Python	>= 3.10

Windows prerequisites

WSL2	
Docker Desktop	>= 4.8.0
Python	>= 3.10

Windows System for Linux (WSL2)

https://learn.microsoft.com/en-us/windows/wsl/install

WSL2 Installation

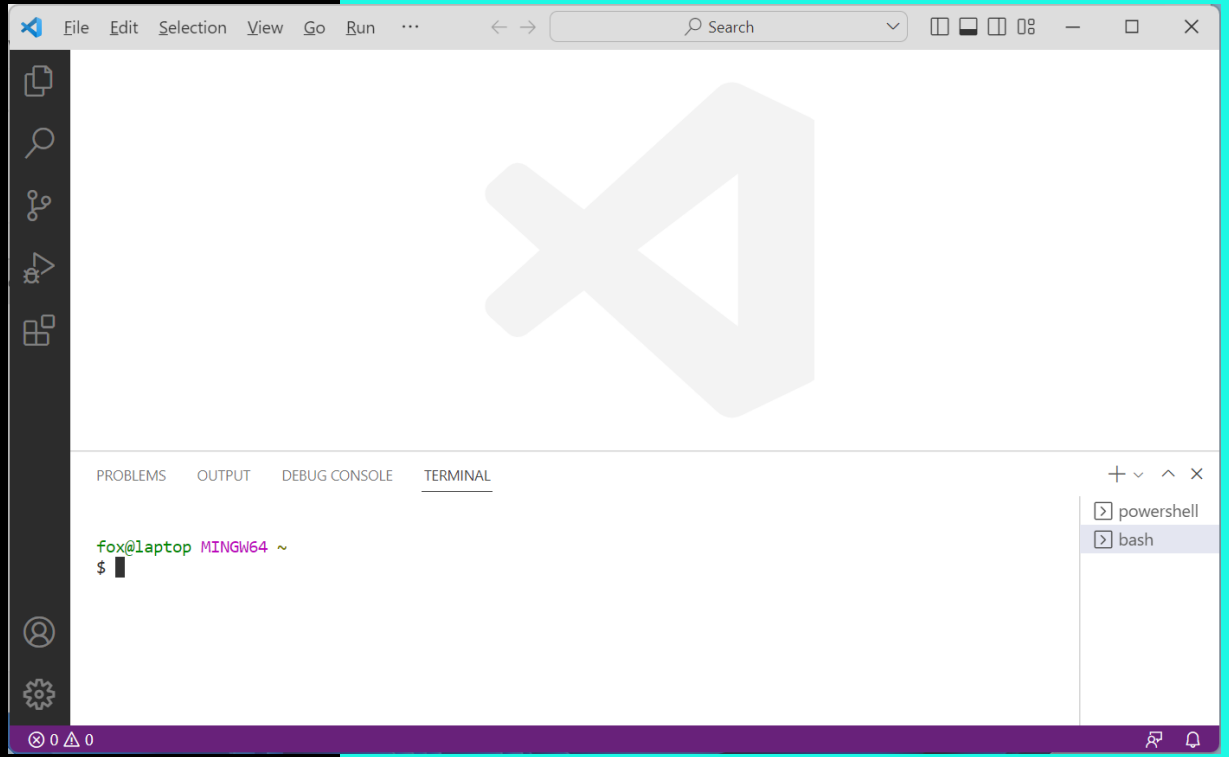
```
> wsl --install
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
Installing: Ubuntu
Ubuntu has been installed.
The requested operation is successful.
```

WSL2 Configuration

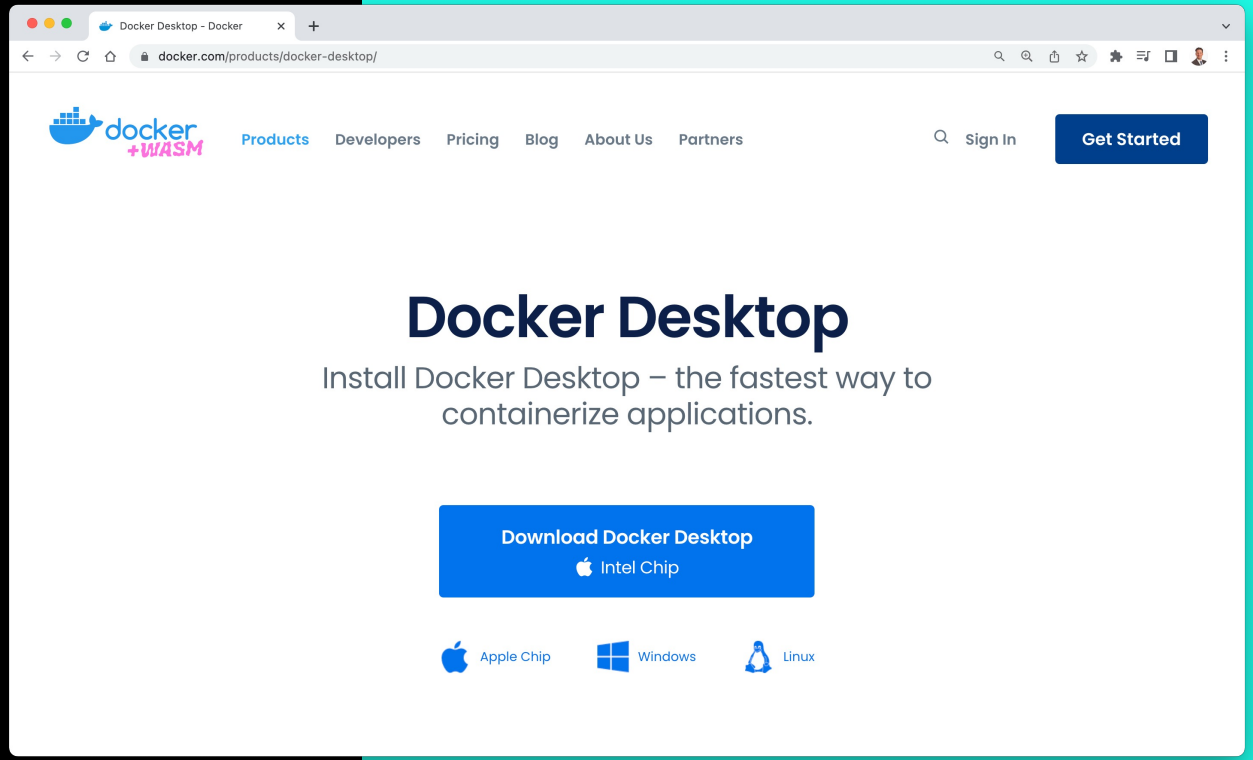
1. Reboot Windows
2. Start Ubuntu
3. Prompted to add **username** and **password**

```
username@hostname:~$
```

Git Bash Installation (for Windows)

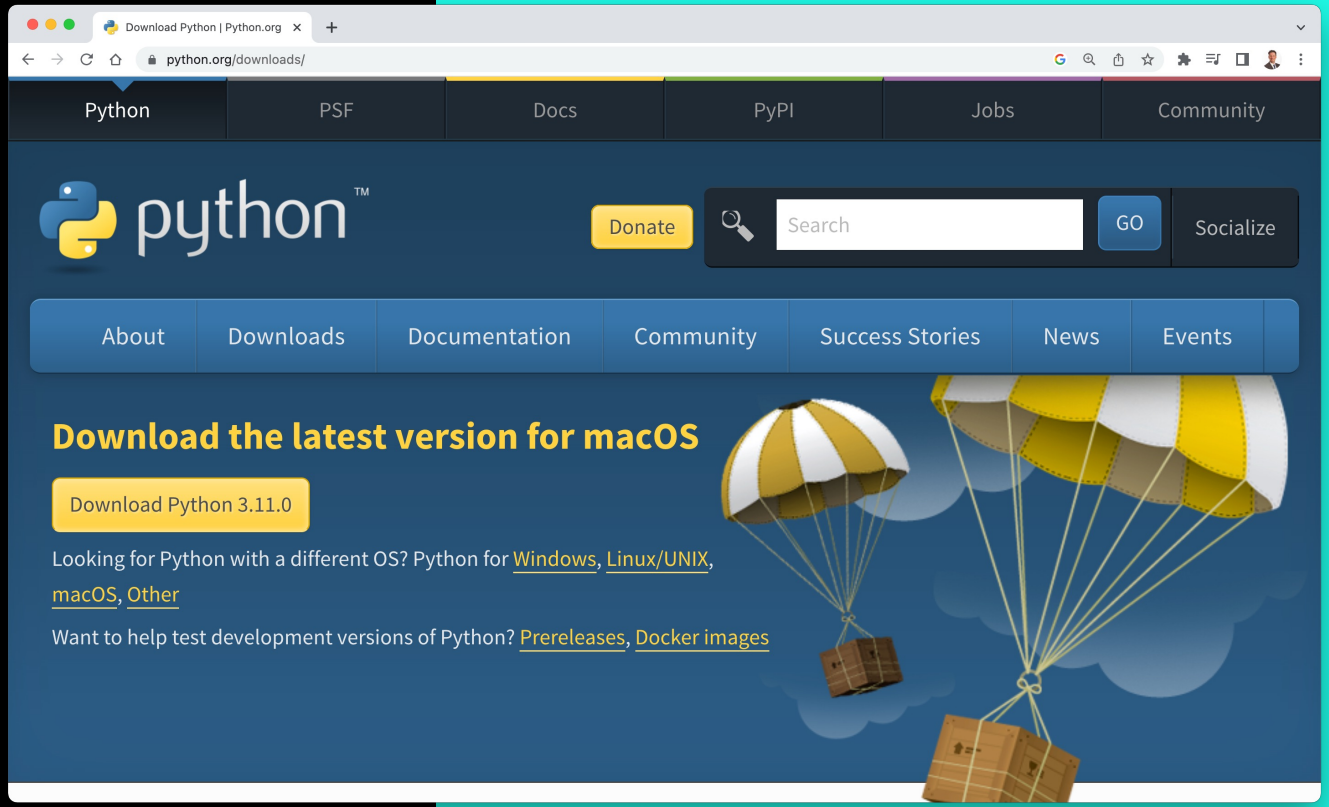


Docker Desktop Installation

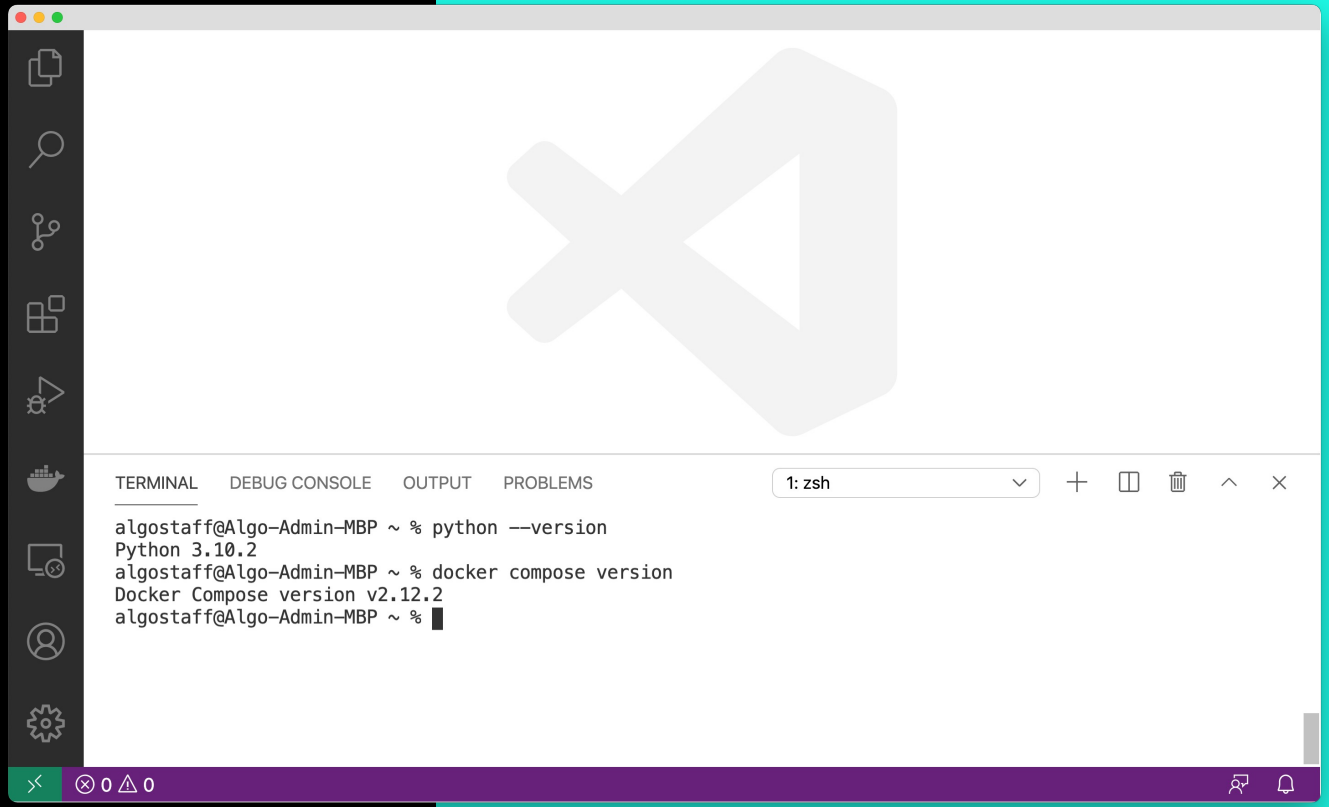


Python Installation

>= 3.10



Verify Prerequisites



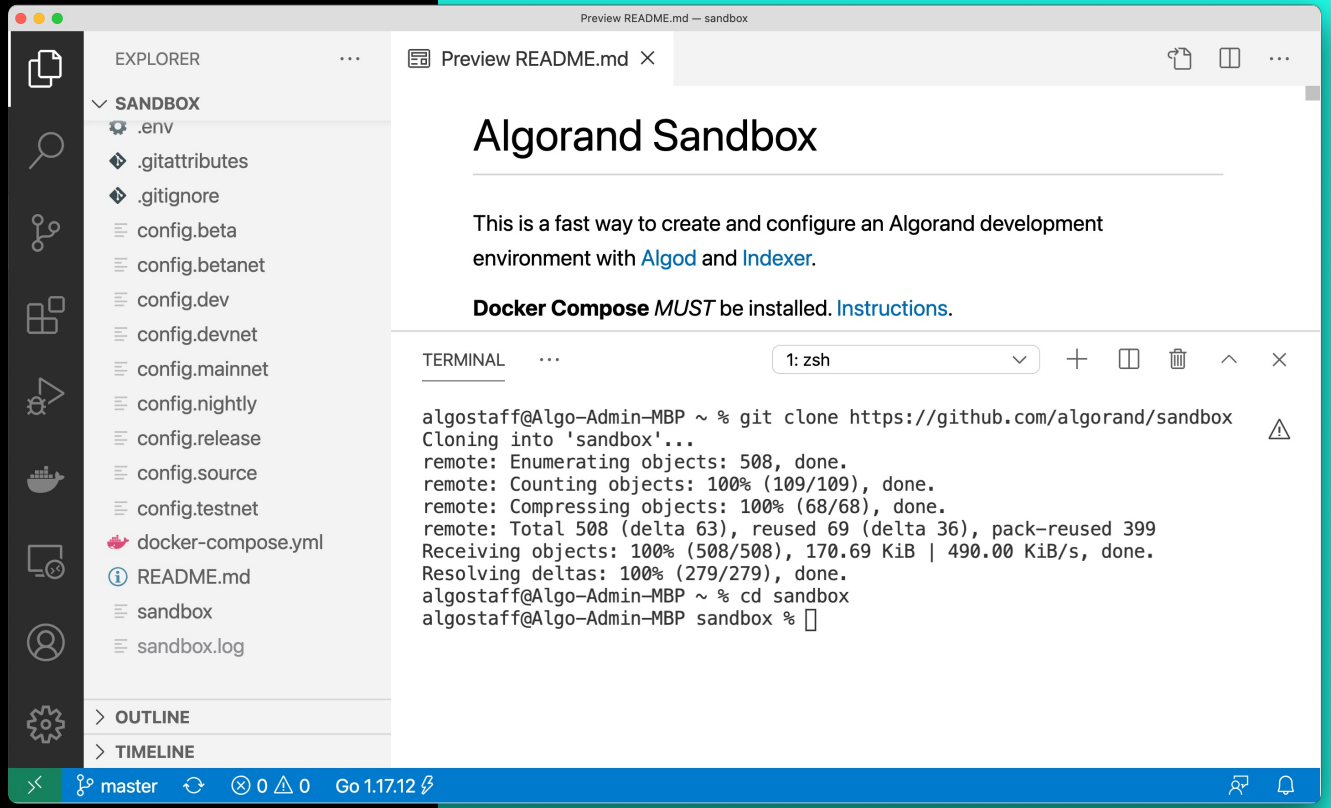
Developer Environment Components

Infrastructure

SDK

Language + Framework

Sandbox Installation



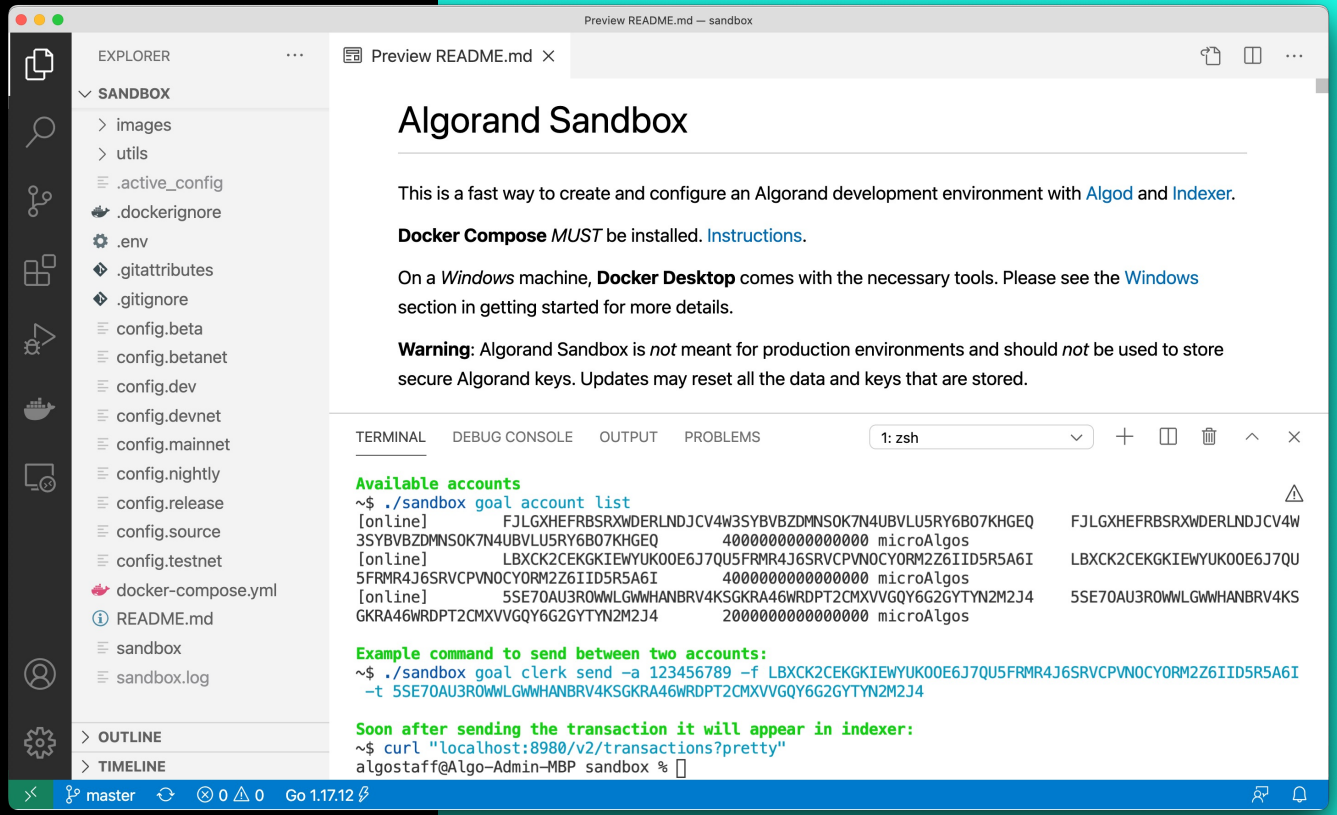
Sandbox Components

Network(s)	4001
Wallet	4002
Query Tool	8980
APIs	ports

Sandbox Execution


```
$ ./sandbox up dev –v.  # start local private net
```

Sandbox Execution



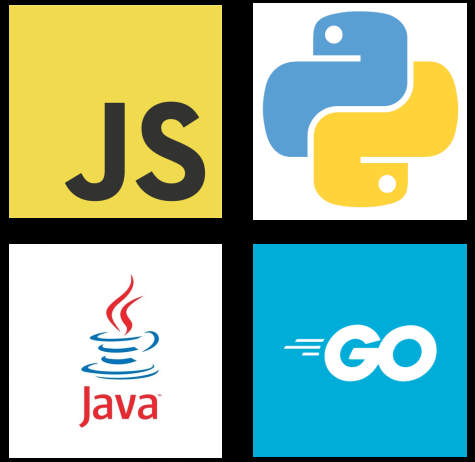
Sandbox Commands


```
$ sandbox help
$ sandbox status
$ sandbox stop
$ sandbox reset
```

Developer Environment Components

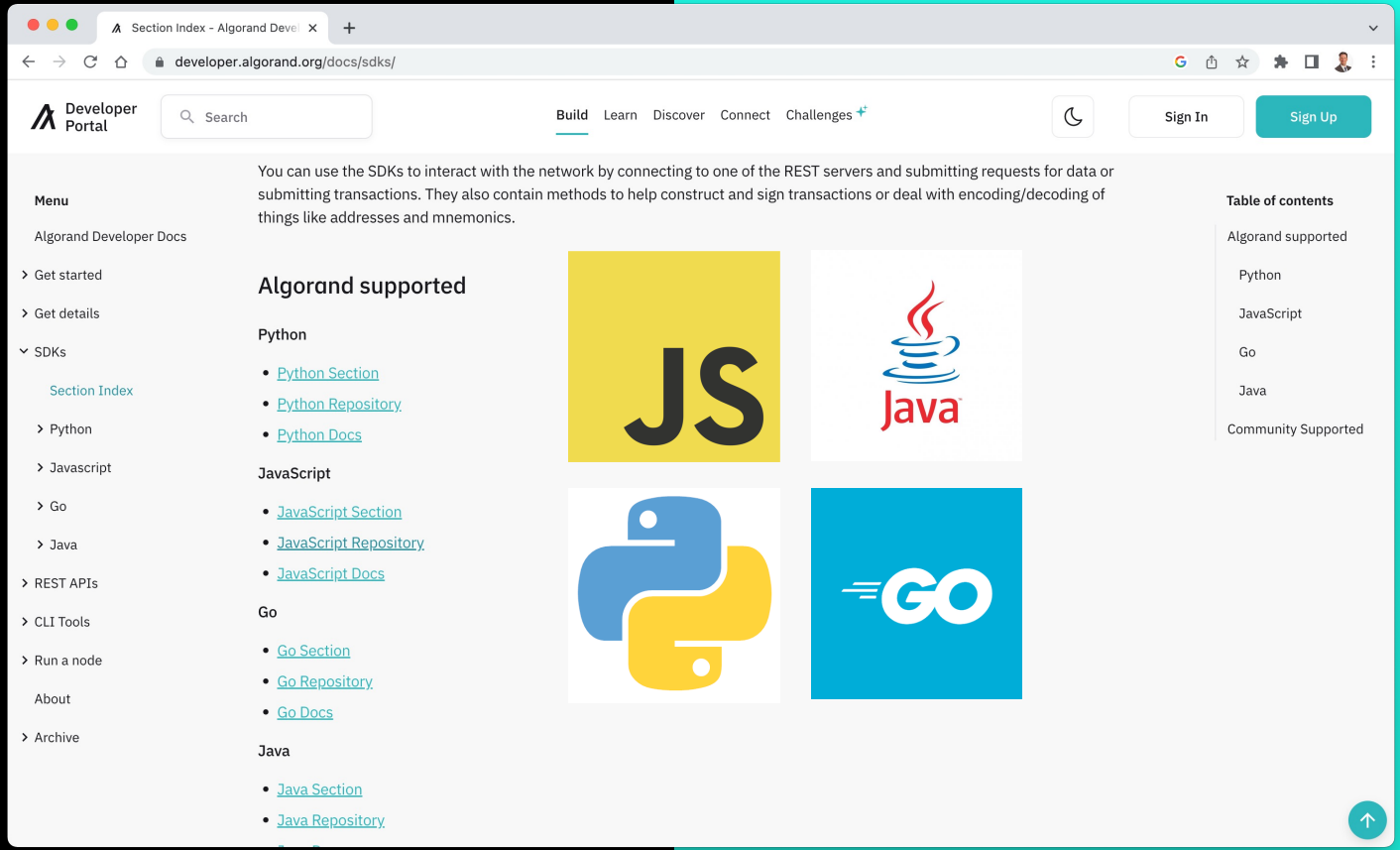
Infrastructure

SDK

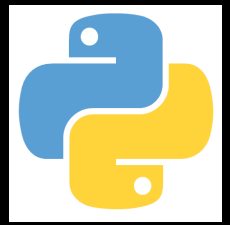


Language + Framework

SDKs


Section Index - Algorand Deve
developer.algorand.org/docs/sdks/
Developer Portal
Search
Build
Learn
Discover
Connect
Challenges
Sign In
Sign Up
Menu
Algorand Developer Docs
Get started
Get details
SDKs
Section Index
Python
Javascript
Go
Java
REST APIs
CLI Tools
Run a node
About
Archive
You can use the SDKs to interact with the network by connecting to one of the REST servers and submitting requests for data or submitting transactions. They also contain methods to help construct and sign transactions or deal with encoding/decoding of things like addresses and mnemonics.
Algorand supported
Python
Python Section
Python Repository
Python Docs
JavaScript
JavaScript Section
JavaScript Repository
JavaScript Docs
Go
Go Section
Go Repository
Go Docs
Java
Java Section
Java Repository
Table of contents
Algorand supported
Python
JavaScript
Go
Java
Community Supported

SDK Installations



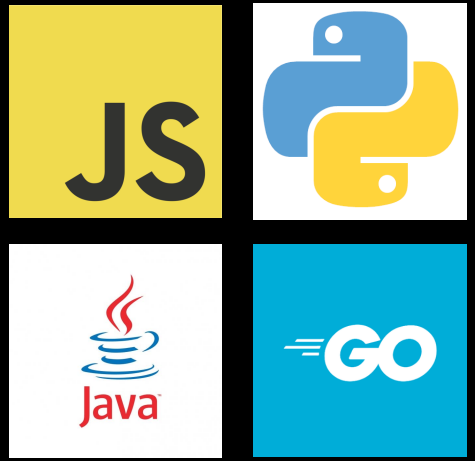
```
$ npm install algosdk
```

```
$ pip install py-algorand-sdk
```

Developer Environment Components

Infrastructure

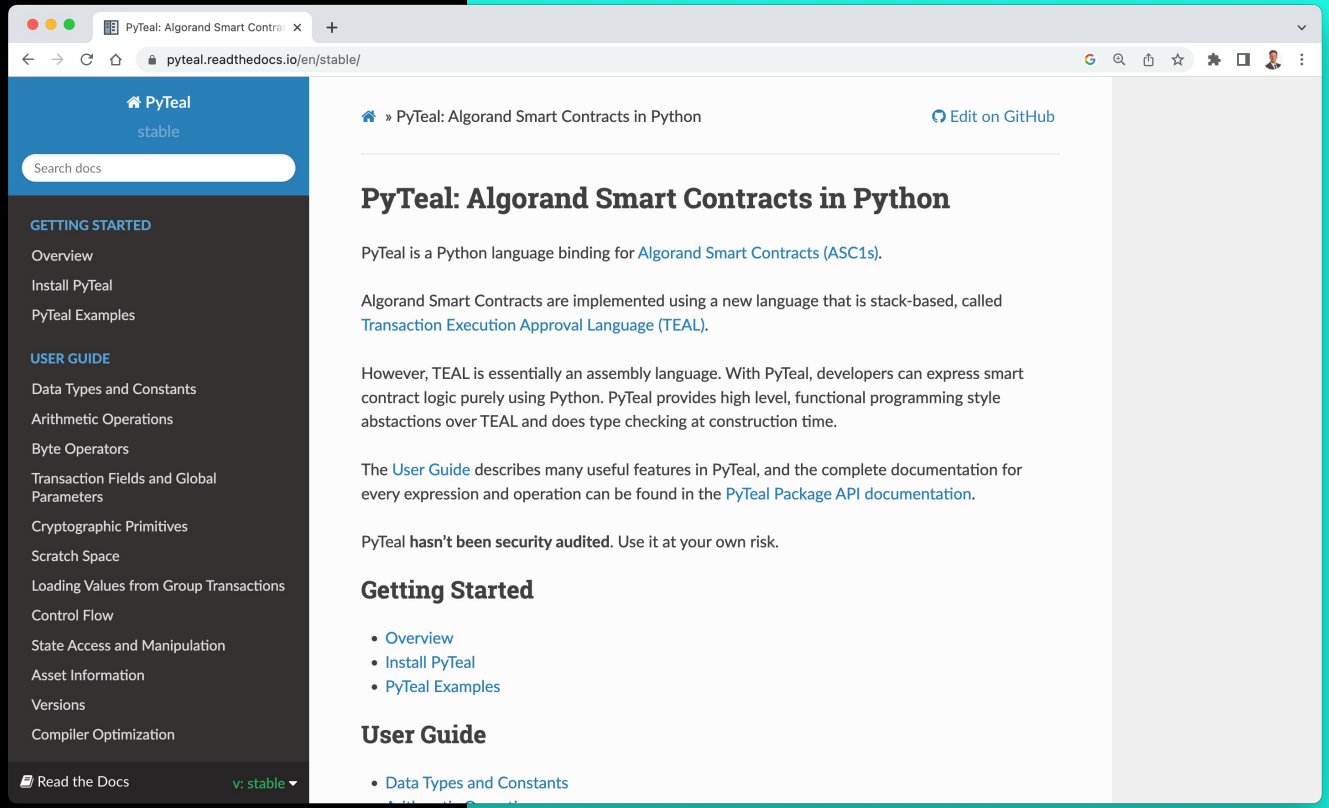
SDK



Language + Framework

Beaker

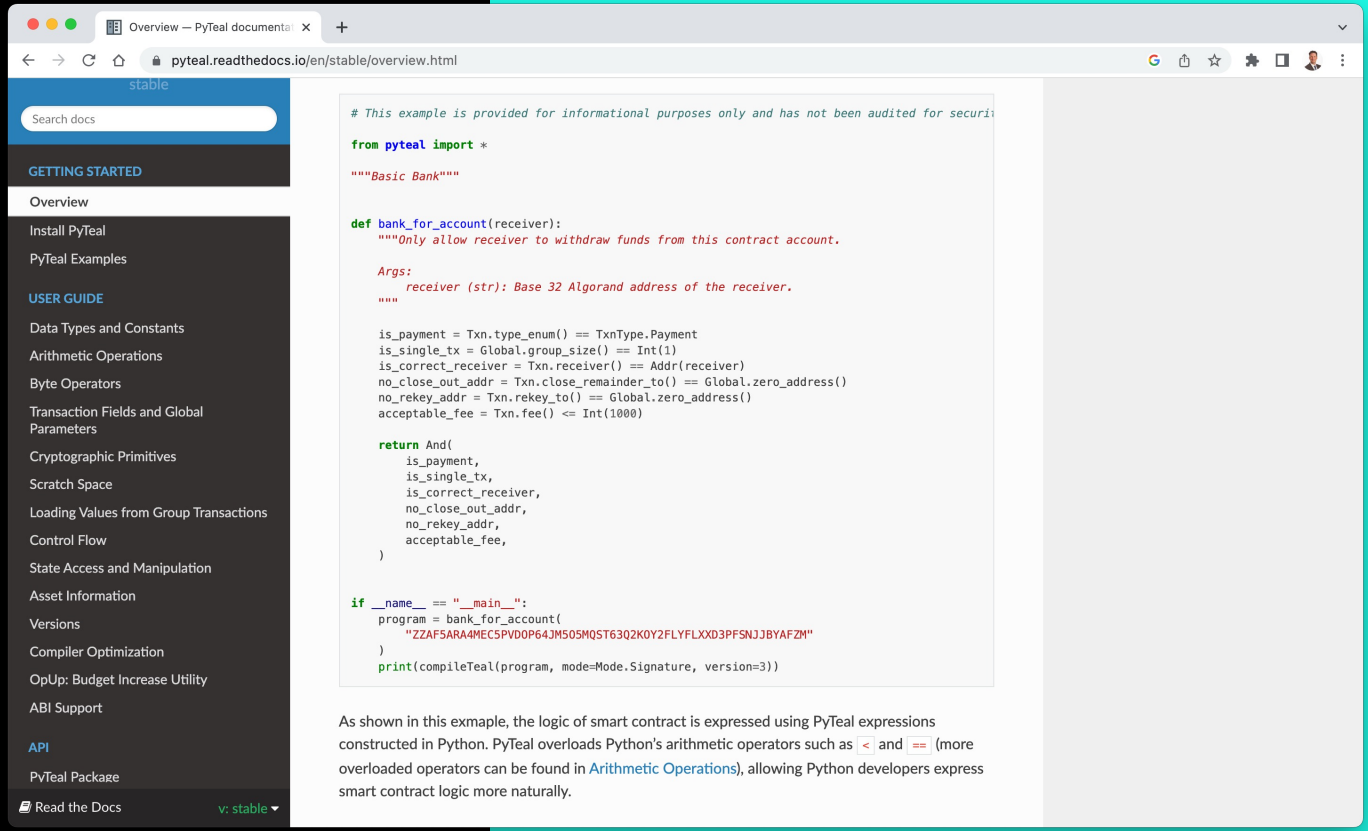
Language


PyTeal: Algorand Smart Contra
pyteal.readthedocs.io/en/stable/
PyTeal
stable
Search docs
GETTING STARTED
Overview
Install PyTeal
PyTeal Examples
USER GUIDE
Data Types and Constants
Arithmetic Operations
Byte Operators
Transaction Fields and Global Parameters
Cryptographic Primitives
Scratch Space
Loading Values from Group Transactions
Control Flow
State Access and Manipulation
Asset Information
Versions
Compiler Optimization
Read the Docs
v: stable
» PyTeal: Algorand Smart Contracts in Python
Edit on GitHub
PyTeal: Algorand Smart Contracts in Python
PyTeal is a Python language binding for Algorand Smart Contracts (ASC1s).
Algorand Smart Contracts are implemented using a new language that is stack-based, called Transaction Execution Approval Language (TEAL).
However, TEAL is essentially an assembly language. With PyTeal, developers can express smart contract logic purely using Python. PyTeal provides high level, functional programming style abstactions over TEAL and does type checking at construction time.
The User Guide describes many useful features in PyTeal, and the complete documentation for every expression and operation can be found in the PyTeal Package API documentation.
PyTeal hasn't been security audited. Use it at your own risk.
Getting Started
Overview
Install PyTeal
PyTeal Examples
User Guide
Data Types and Constants

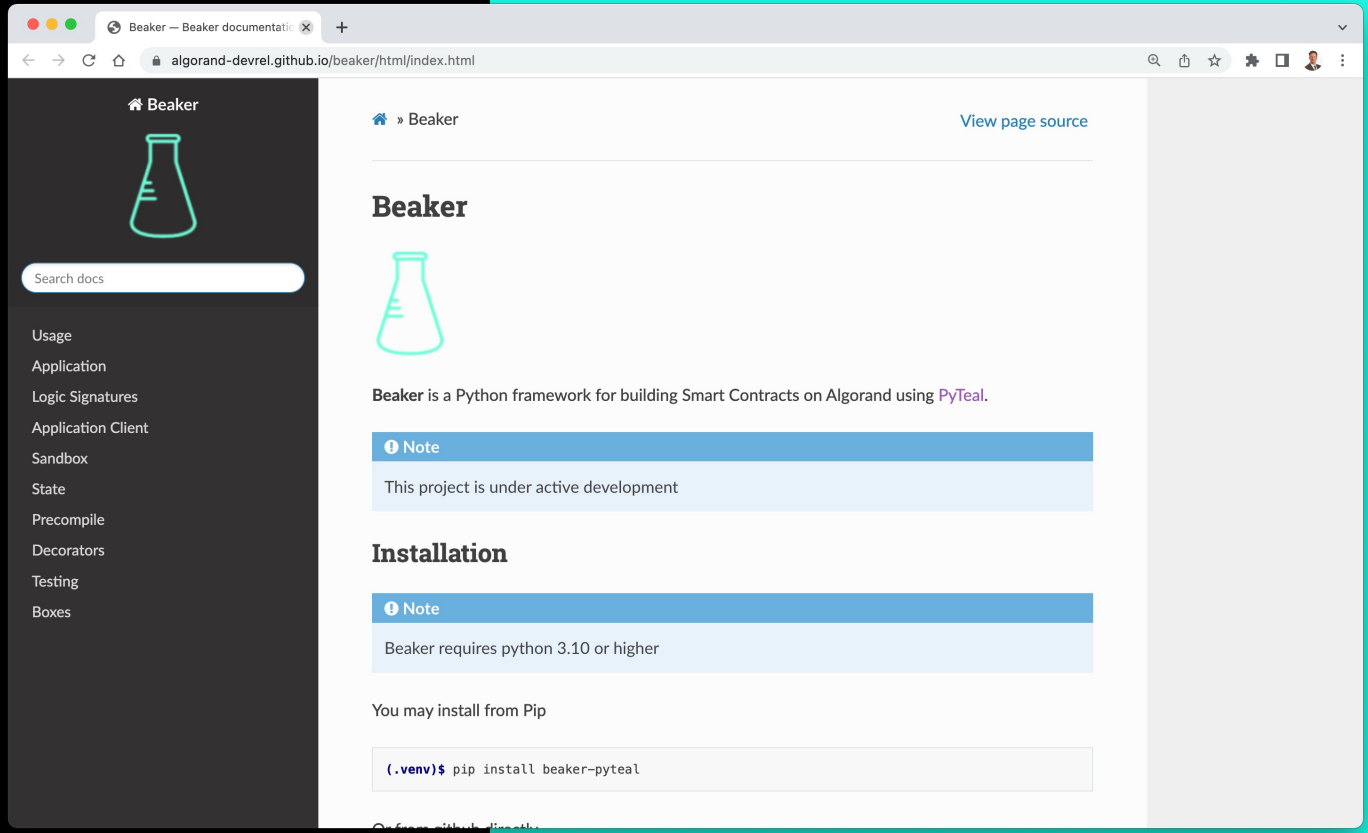
PyTeal Installation


```
$ pip install pyteal
```

PyTeal Syntax



Framework


Beaker — Beaker documentatic
algorand-devrel.github.io/beaker/html/index.html
Beaker
Search docs
Usage
Application
Logic Signatures
Application Client
Sandbox
State
Precompile
Decorators
Testing
Boxes
» Beaker
View page source
Beaker
Beaker is a Python framework for building Smart Contracts on Algorand using PyTeal.
Note
This project is under active development
Installation
Note
Beaker requires python 3.10 or higher
You may install from Pip
(.venv)$ pip install beaker-pyteal

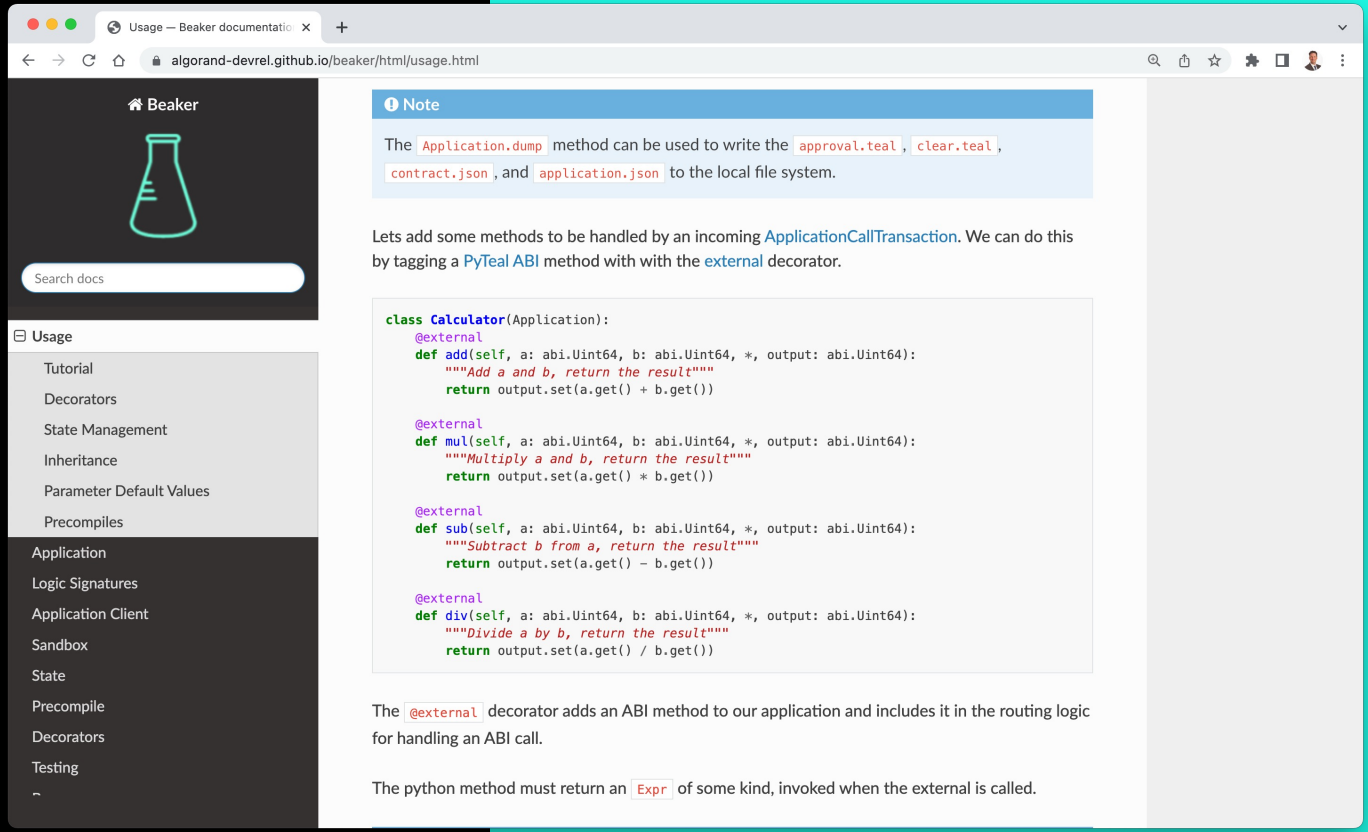
Beaker Installation

Beaker

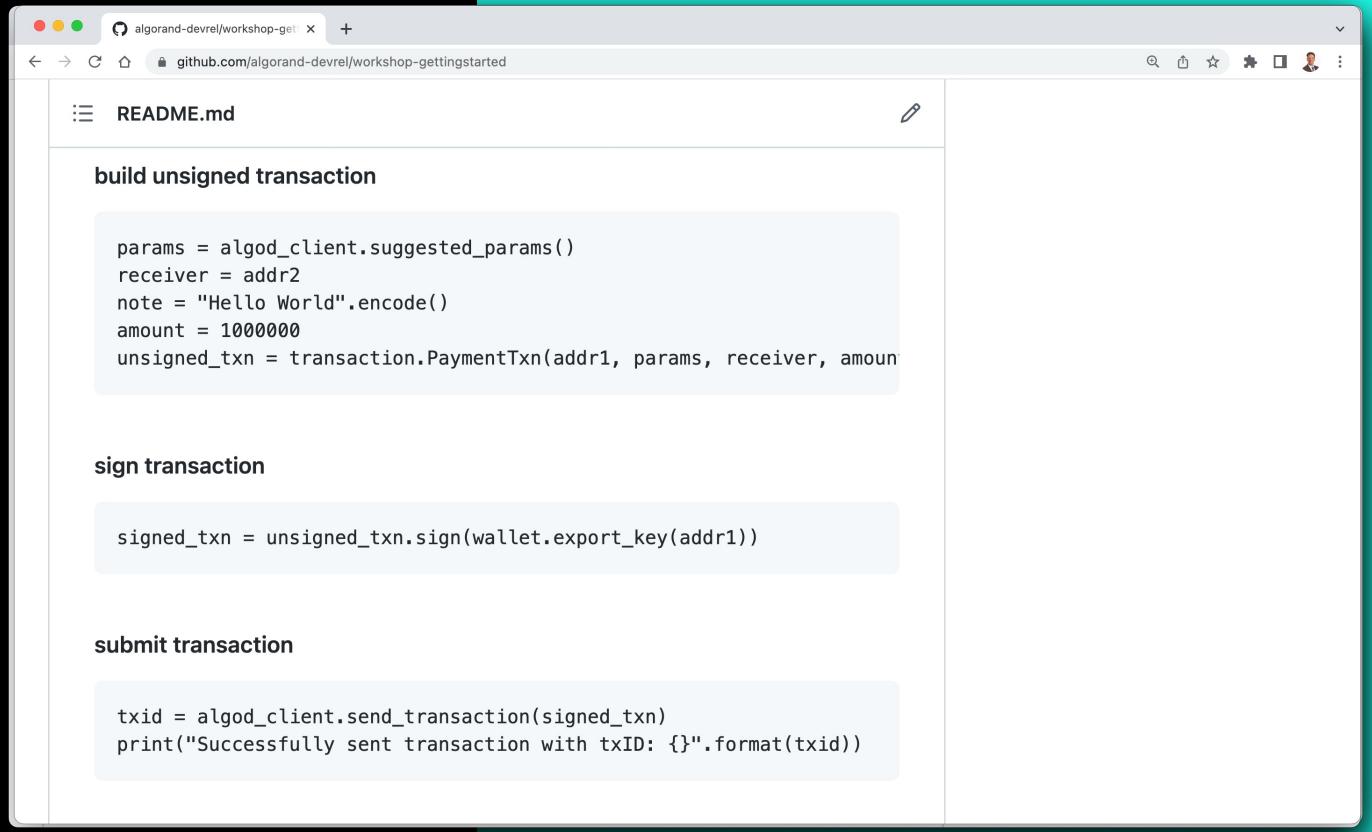
(auto installs pyteal)

```
$ pip install beaker-pyteal
```

Beaker Syntax



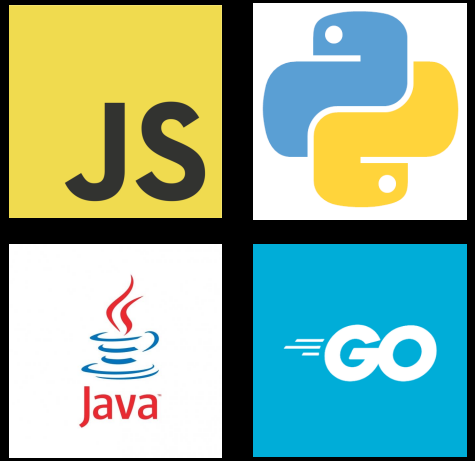
Getting Started



Developer Environment Components

Infrastructure

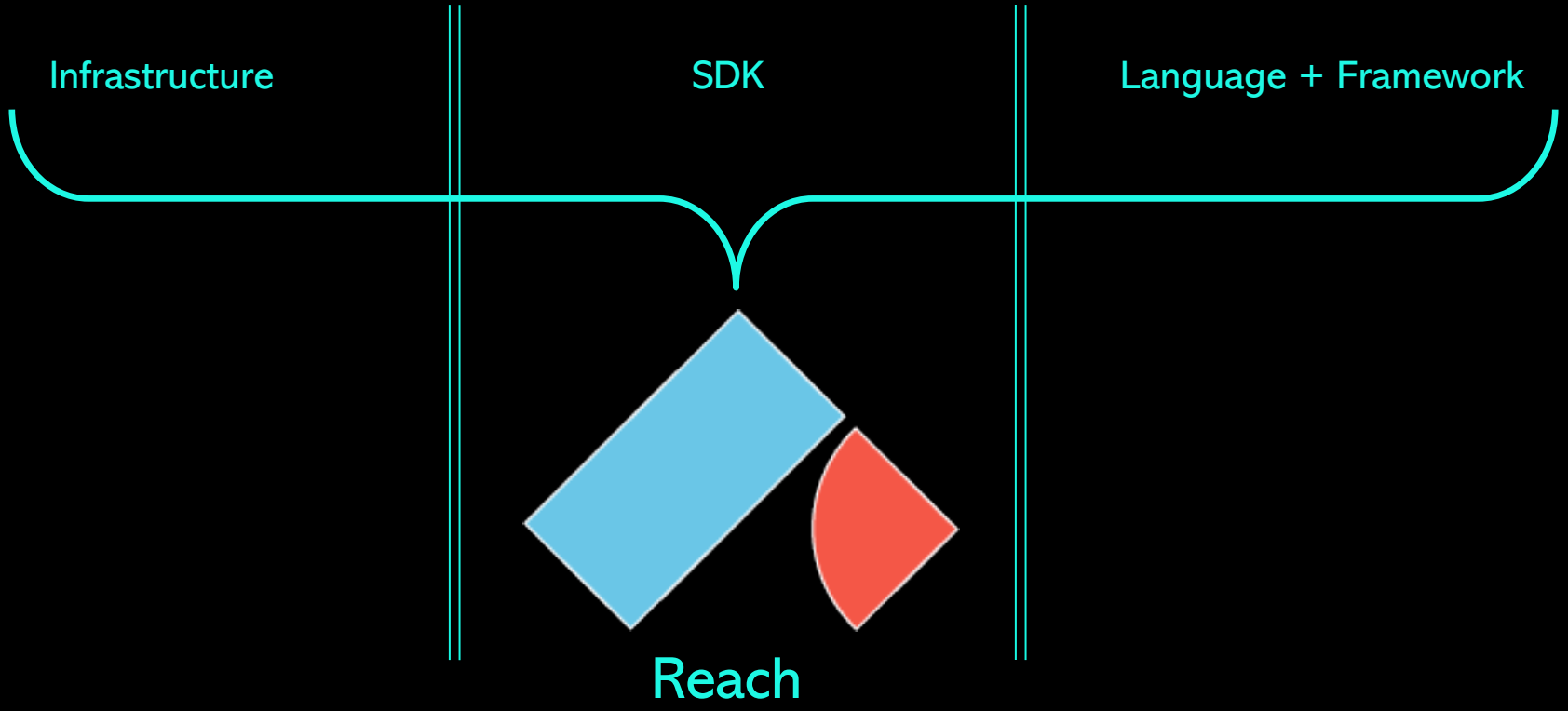
SDK



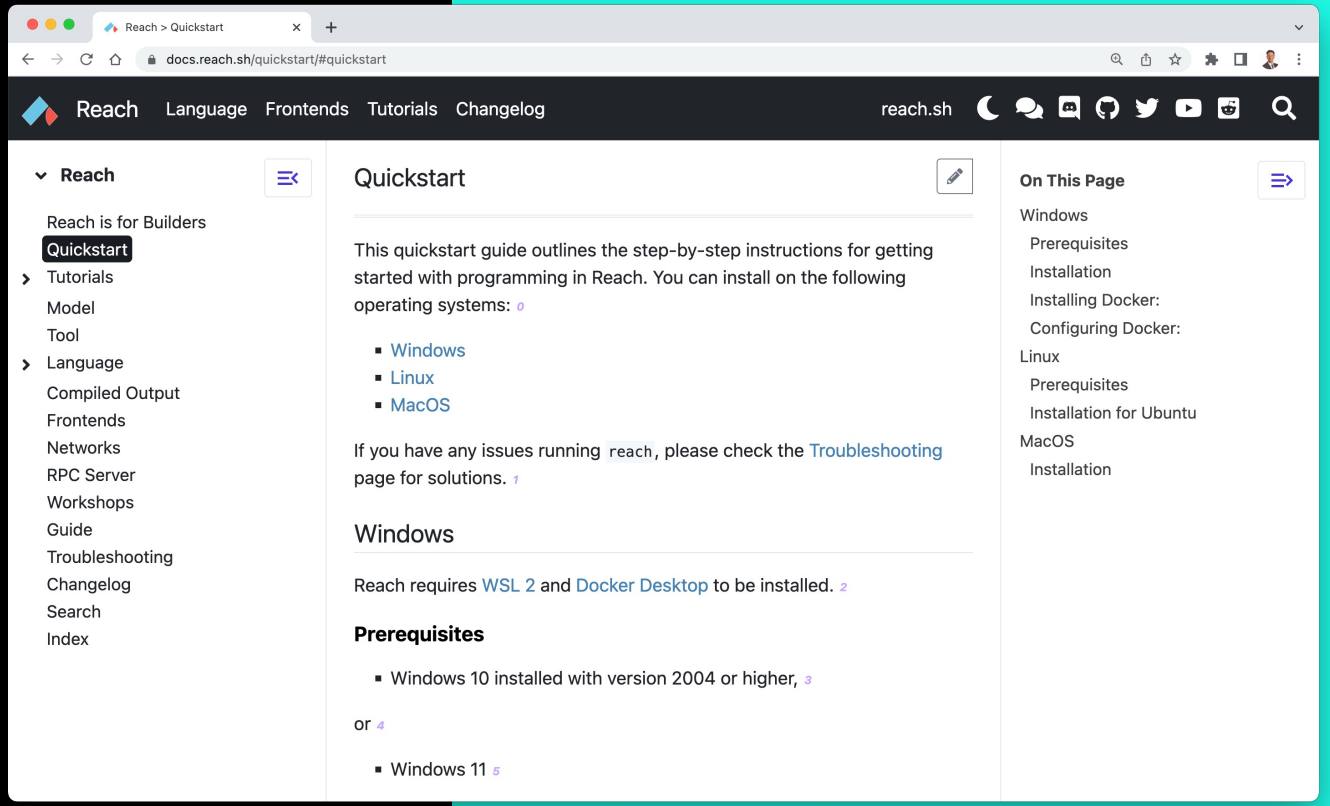
Language + Framework

Beaker

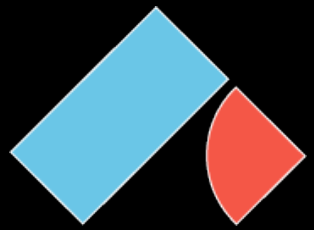
Developer Environment Components



Reach Installation

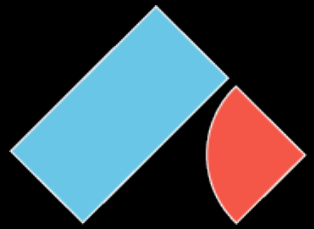


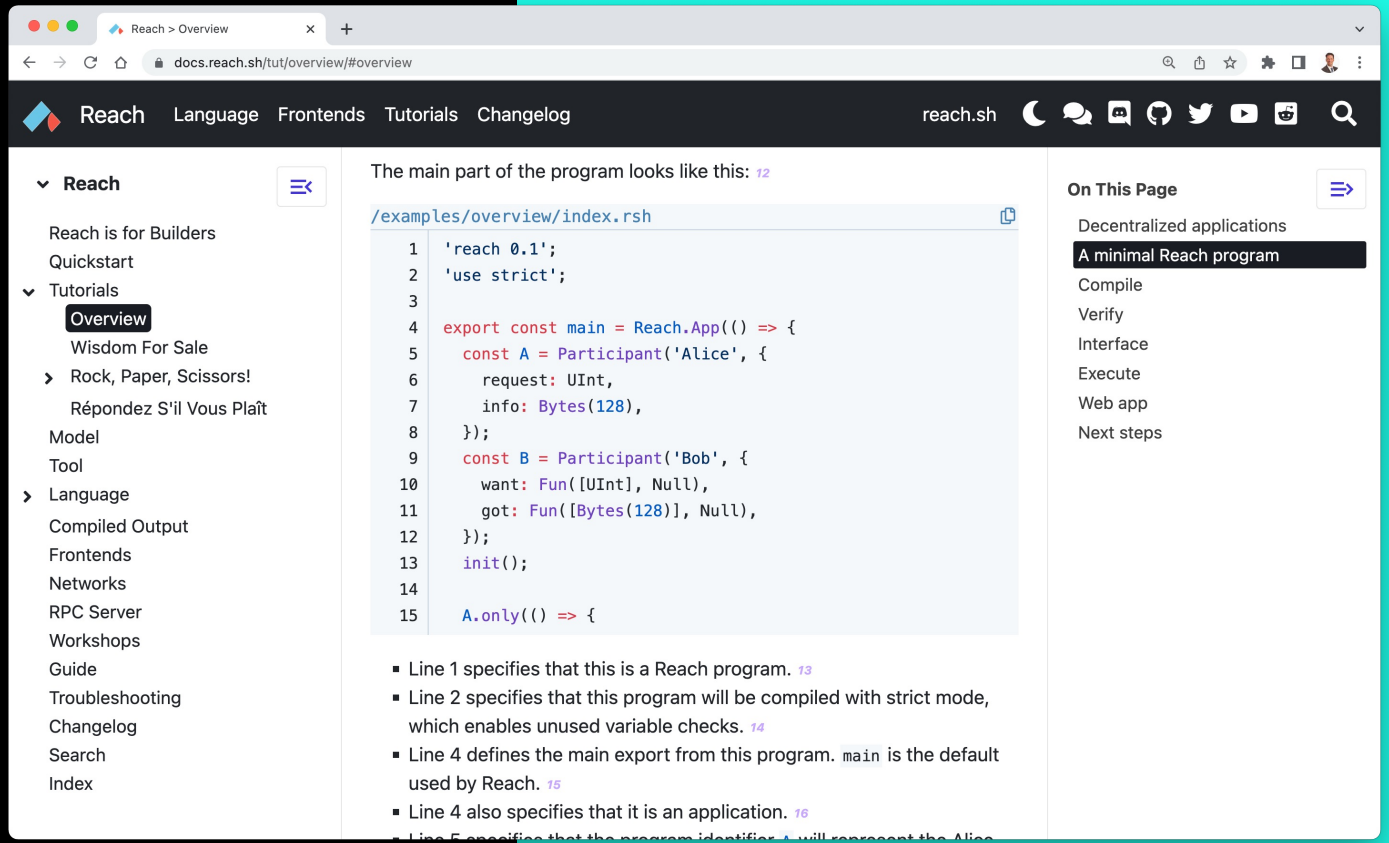
Reach Installation



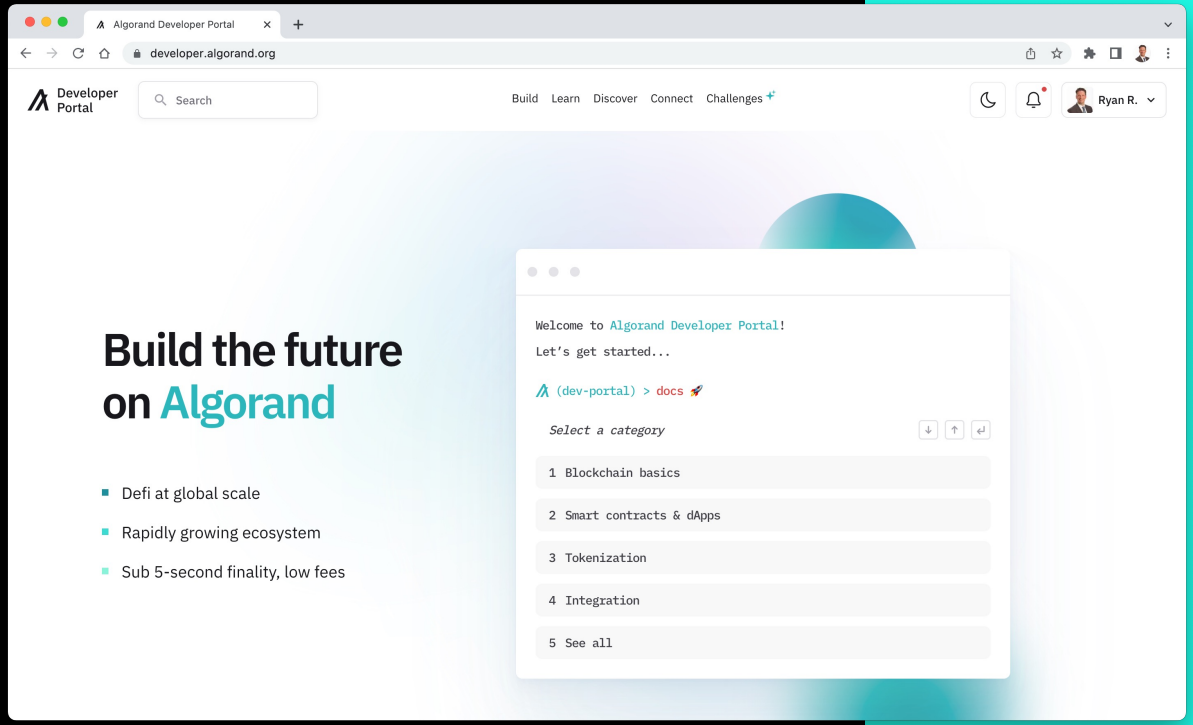
```
$ sudo apt install make curl
$ mkdir -p ~/reach && cd ~/reach
$ curl https://docs.reach.sh/reach -o reach ; chmod +x reach
$ ./reach version
```

Reach Syntax



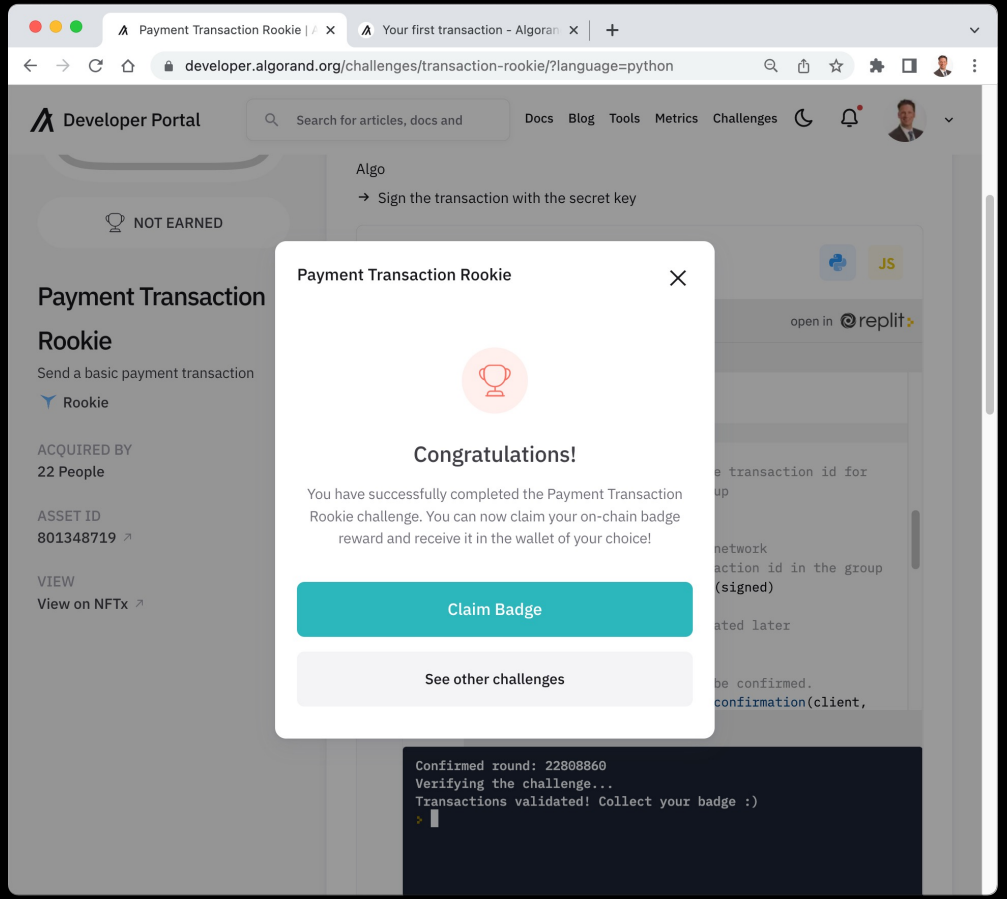


Developer Portal



https://developer.algorand.org/

Developer Portal



Show me the code

#THANKS

@ryanRfox

Ryan R. Fox

Developer Advocate

@Algorand

Let's Connect