

Week 4 of 6

# 4

# Lending, borrowing & stablecoins

**Task**

Model a loan to liquidation in a spreadsheet, with a live Pyth price

**Blog post**

Understanding lending and borrowing in DeFi

# Today

**1** A bank with no banker

---

**2** The four numbers

---

**3** Liquidation, step by step

---

**4** Where prices come from

---

**5** Stablecoins: the dollars of DeFi

---

---

# **A bank with no banker**

Lenders, borrowers, and a program in the middle.

# No name, no credit score, no court

---

## A bank lends on

- who you are
- your income and history
- the right to sue you

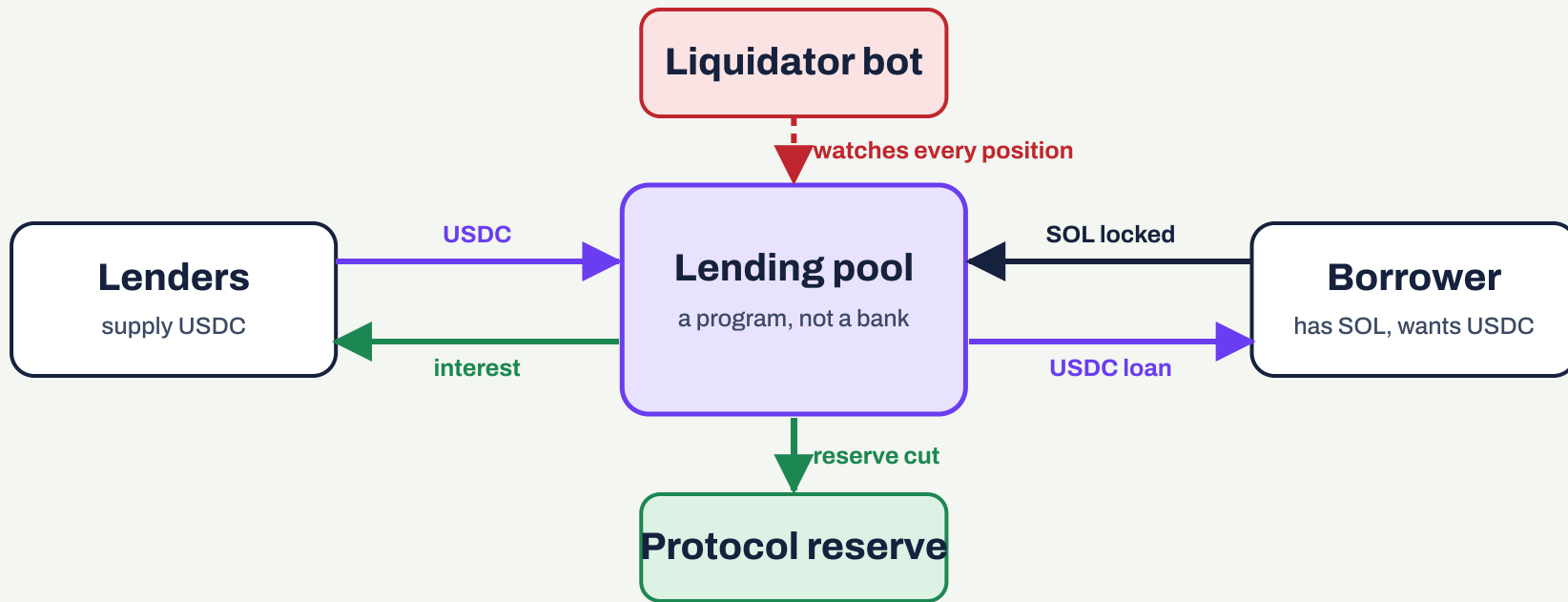
## A lending program knows

- an address
- what that address locked up

So every loan is worth less than its collateral.

**\$150 in → borrow at most \$105**

# Who pays whom



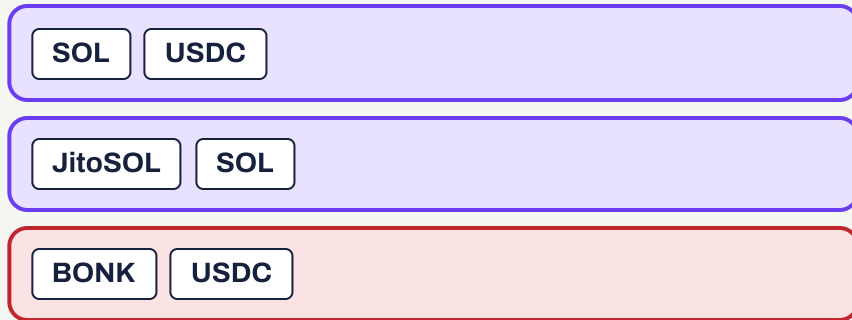
# Solana's lenders

## Shared pool



Every asset backs every loan. One bad asset can hurt everyone.

## Isolated markets



Each market stands alone. A bad asset only hurts its own market.

[Kamino Lend](#) · [Jupiter Lend](#) · [Save](#) · [Project 0](#)



# The four numbers

LTV, liquidation threshold, health factor, utilization.

# One position, one bar



LTV

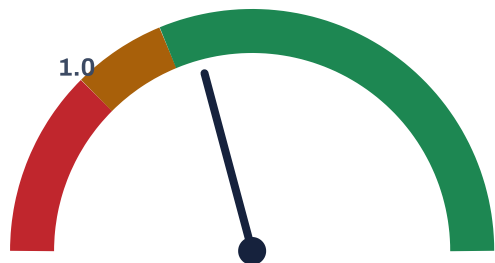
$$900 \div 1,500 = 60\%$$

Health factor

$$1,500 \times 0.80 \div 900 = 1.33$$

Below 1.00 = liquidation

# Your position, live



**1.33**

**Healthy**

SOL price **\$150**



USDC borrowed **\$900**



Collateral

**10 SOL = \$1,500**

LTV

**60.0%**

Liquidation price

**\$112.50**

Can still borrow

**\$150**

$HF = \text{collateral} \times 0.80 \div \text{debt}$



# Liquidation

What happens at a health factor below 1.

# The walkthrough



Collateral

**5.705 SOL = \$627.55**

Debt

**\$450**

LTV

**71.7%**

Health

**1.12**

## Liquidator bot

Repays **450 USDC** of debt (50% close factor)

Takes  $\$450 + 5\% \text{ bonus} = \$472.50$  of **SOL** = 4.295 SOL

Profit: **\$22.50**, before fees

## Borrower after

Lost 4.295 SOL. Still owes \$450. Health back above 1, at **1.12**.

The 5% bonus is the penalty they paid.

# Flash loans: a million, for one transaction

One transaction, all or nothing

1

Borrow **1,000,000 USDC**

no collateral

2

Use it

repay someone's debt in a liquidation, or buy low here, sell high there

3

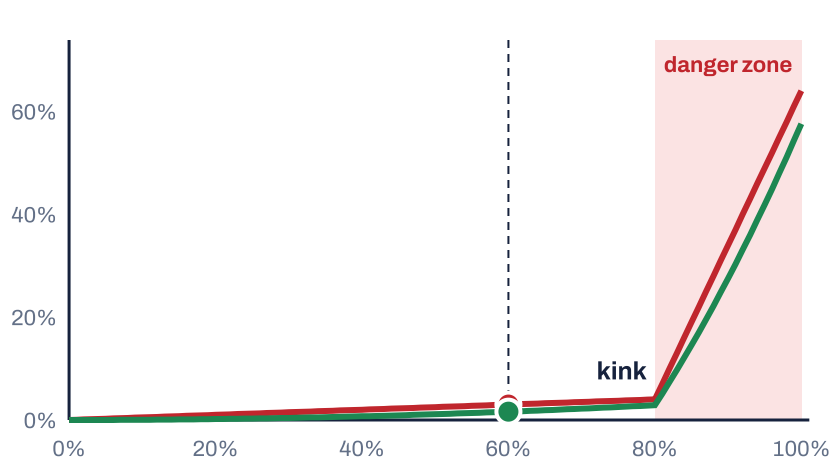
**Can't repay**

the program refuses to finish

## Reverted

As if steps 1 and 2 never happened. Only the network fee is spent.

# What borrowing costs



Utilization 60%



utilization = borrowed ÷ supplied

— Borrowers pay

**3.0%**

— Lenders earn

**1.6%**

Plenty of cash in the pool. Rates stay low.

**10 SOL collateral, 900 USDC debt, liquidation threshold 80%. At what SOL price can you be liquidated?**

**A** \$90.00

**B** \$112.50

**C** \$128.57

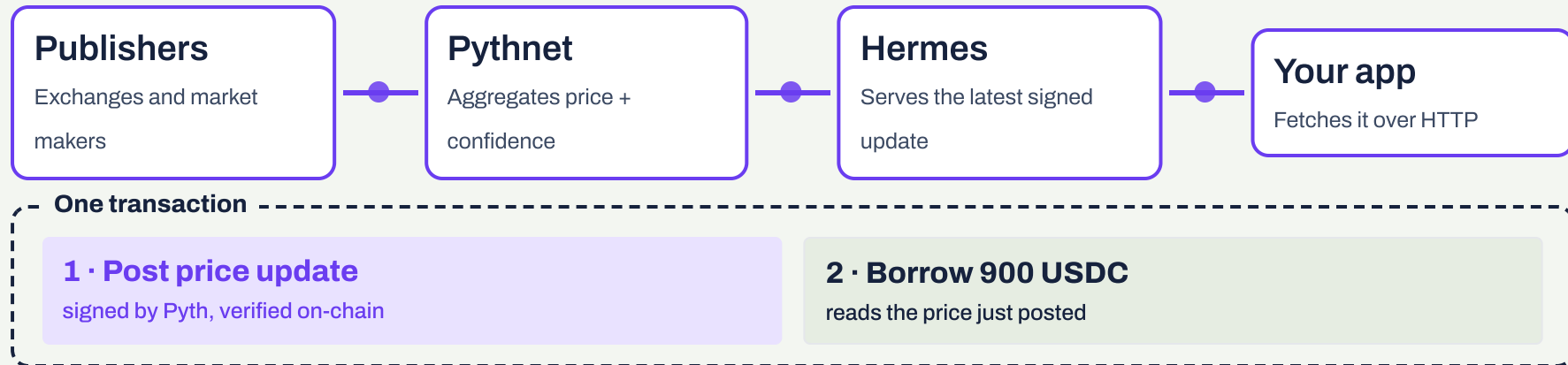
**D** \$150.00



# **Where prices come from**

A lender is only as safe as its price feed.

# How Pyth delivers a price



Program checks: **signature valid** · **feed is SOL/USD** · **not too old**, then prices your collateral.

---

price: 15234000000, expo: -8 →  $15234000000 \times 10^{-8} = \text{\$152.34}$

# See the live price

## Users

```
insights.pyth.network
```

Search **SOL/USD**. Read the price, the confidence band and when it last updated.

## Builders

```
npm run w4:price
```

Reads Pyth's SOL/USD account on Solana: **price** × 10<sup>expo</sup>, after checking who owns the account.

### Hermes now needs a key

Since 26 Aug 2026 the public Hermes URL returns 401. Apps use a key; you don't need one for class.

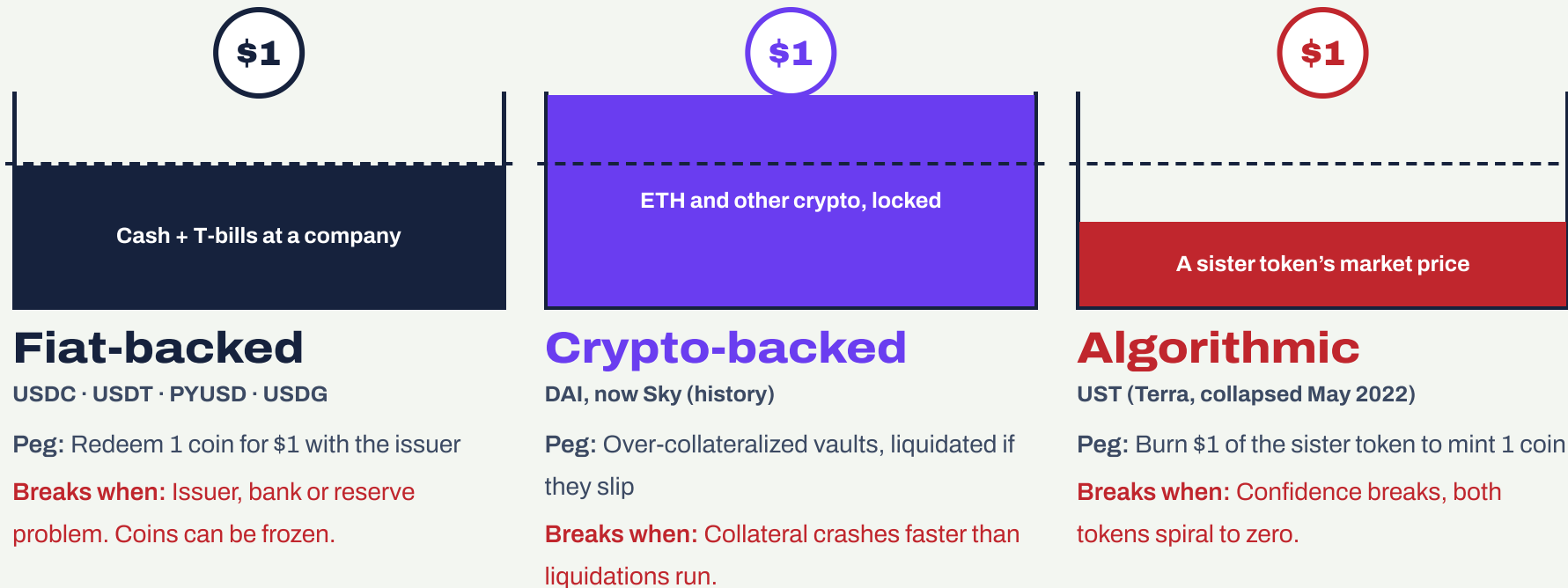
Write down **the price and the time you looked**. That's your spreadsheet's starting row.



# **Stablecoins**

The dollars of DeFi.

# Three ways to hold \$1



# What holds the peg



Market price \$0.970



☒ Issuer is redeeming

**Buy at \$0.970, redeem for \$1.00**

Profit 3.0¢ per coin. Buying lifts the price; redeeming burns supply.

# Why it matters here

## Payments

Settles in under a second for a fraction of a cent, any day of the week.

## Remittance

Wallet to wallet across borders, no correspondent bank.

## Savings

A dollar balance without a dollar bank account.

## The catch

Off-ramps, local rules, issuer freezes, and scams.

**USDC trades at \$0.96 and Circle is redeeming normally. What pulls it back to \$1?**

**A** Circle prints more USDC

**B** Traders buy at \$0.96 and redeem for \$1

**C** A Solana validator vote

**D** Nothing, it stays there

# Task: take a loan to liquidation

## Users

### Simulate it (recommended)

1. Get the live SOL price (insights.pyth.network)
2. Spreadsheet: deposit, borrow, then drop the price in steps
3. Each row: LTV, health factor
4. Mark the liquidation price and the liquidator's profit

## Builders

### Mainnet fork

1. `surfpool start` : real Kamino and Jupiter Lend accounts, locally
2. Fund a wallet with cheatcodes
3. Deposit and borrow through the SDK
4. Move time and price, watch health

### Optional, real money

Already hold USDC? Supply \$5 to a lender and watch interest accrue. Supply only. Never borrow.

## **Blog post**

# **"Understanding lending and borrowing in DeFi"**

Include your liquidation walkthrough, and one paragraph on which stablecoin you'd trust, and why.



# Teacher prep

For you, before class. Skip this live.

# You, before class

1. **Learn the four numbers.** LTV, liquidation threshold, health factor, utilization. Then the kinked rate curve.
2. **Walk one liquidation by hand.** 10 SOL at \$150, borrow 900 (60%). SOL to \$110:  $LTV \approx 82\%$ .
3. **Look up SOL's real liquidation threshold** on Kamino or Jupiter Lend. Who gets liquidated, and how does the liquidator profit?
4. **Read Pyth's Solana pull-integration guide.** Week 5 depends on it.
5. **Know the stablecoin types**, and which have real depth on Solana.

# Verify before class

## Any devnet lending market

No lender has a public devnet UI. Builders: create a Kamino devnet market with klend-sdk, or fork mainnet with Surfpool.

## Numbers with dates

Stablecoin supply (\$17.3B, 25 Sep 2026) and TVL move weekly. Re-check on DefiLlama the day you teach.

## marginfi is now Project 0

Same team and program, renamed Sep 2025. Old app shut in mid-2026; use app.0.xyz.

## Builders' fork

`surfpool start` · port 8899 · cheatcodes for balances and time