Polymarket bots are clearing six figures a month with this exact strategy.

Most people trade based on intuition. They trade based on **latency**.

This bot waits for the literal last second before a market settles. It detects assets guaranteed to settle at $1.00 but trading at $0.98 due to panic or liquidity gaps.

- **Risk:** Near Zero (Latency Arbitrage).
- **Strategy:** 15-minute expiration sniping.
- **Result:** Picking up free money in front of a steamroller.

I'm going to show you how to rebuild this engine from scratch—no coding experience required.

Part 1: The Stack (What you need)

You don't need to be a developer, but you need the right tools.

1. Install VS Code. This is your command center.

- Download: code.visualstudio.com
- *Tip: Install the "Python" extension inside VS Code.*

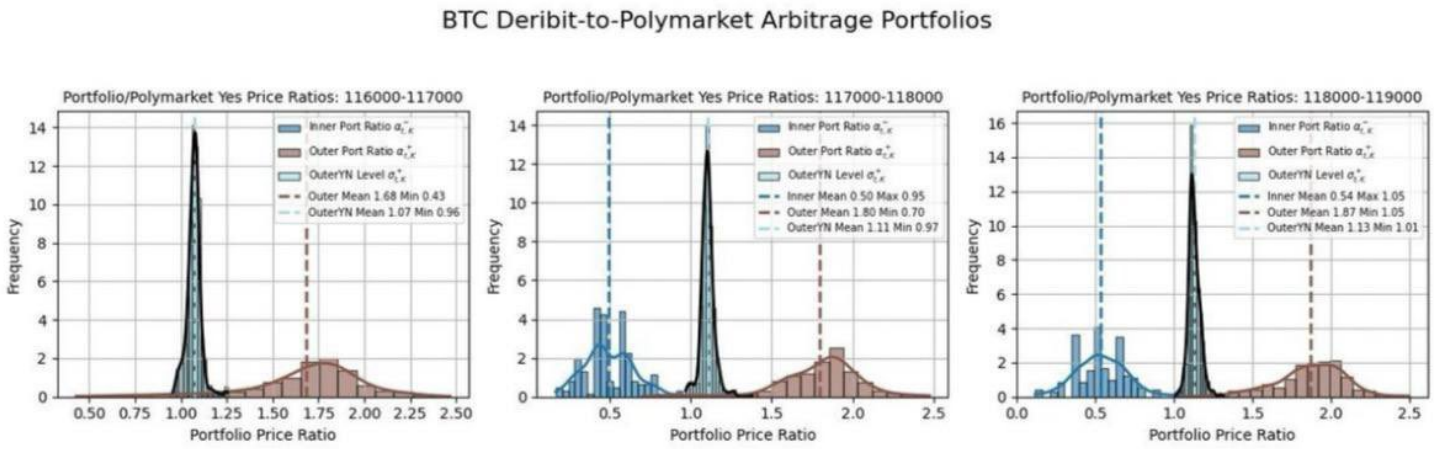
2. Install Python

- Download Python 3.10+ from python.org.
- ⚠️ **CRITICAL:** During installation, check the box that says **"Add Python to PATH"**. If you miss this, nothing works.

3. The Secret Sauce (Libraries) Open your terminal (Command Prompt or Terminal) and run this single command to install the tools used by Polymarket pros:

Bash

pip install asyncio aiohttp websockets python-dotenv py-clob-client



Part 2: The Access (Keys)

The bot needs permission to trade.

1. **Get Your API Keys:** Go to Polymarket → Settings → API. Create a key. Save the Key, Secret, and Passphrase. *You also need your wallet's Private Key (Export from Metamask/Phantom).*
2. **Create the Safe:** Open VS Code. Create a new file named .env. Paste your keys in this exact format:Code snippetPRIVATE_KEY=0x... (your wallet private key) POLYGON_CHAIN_ID=137 CLOB_API_KEY=... CLOB_SECRET=... CLOB_PASSPHRASE=...

Part 3: The "No-Code" Prompts

We aren't writing code. We are prompting AI (ChatGPT o1, Claude 3.5, or Gemini) to write it for us. Copy these exactly.

Step A: The Allowance Script (Required)

You must authorize the exchange to spend your USDC or the bot will fail.

Prompt to generate approve.py:

"Write a Python script using py-clob-client to approve USDC for trading on Polymarket. Load PRIVATE_KEY from a .env file. Initialize the

ClobClient on Polygon Mainnet (Chain ID 137). Call client.set_allowance to authorize the exchange to spend my USDC. Print 'Success' or the specific error message."

Step B: The Market Scanner

This script finds the active 15-minute window so you don't have to search manually.

Prompt to generate scanner.py:
"Write a Python script using aiohttp to query the Polymarket Gamma API. **Goal:** Find the currently active 15-minute Bitcoin or Ethereum market. **Logic:** Get current time in ET. Calculate the current 15-minute window (e.g., 10:00–10:15). Search https://gamma-api.polymarket.com/public-search for 'Bitcoin Up or Down' matching this window. Return the condition_id, token_id for YES/NO, and the end_time. Filter for markets ending in less than 20 minutes."

Step C: The Sniper Building

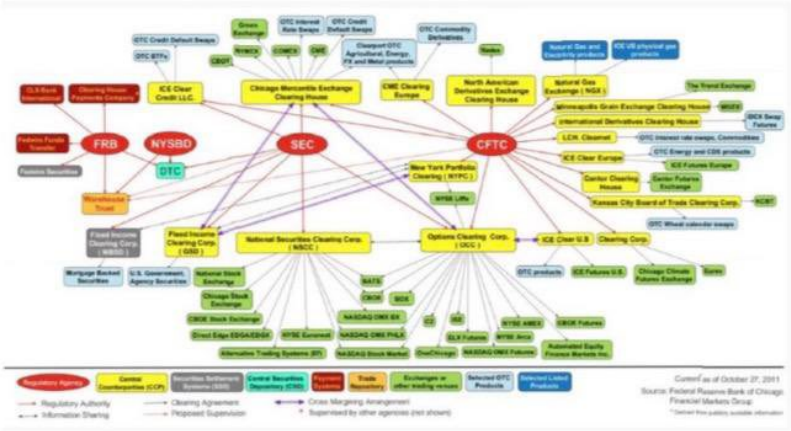
This is the money printer.

Prompt to generate sniper.py:
"Write a high-frequency trading script using websockets and py-clob-client. **Context:** I have a target Condition ID and Token IDs. **The Logic:** **Setup:** Connect to wss://ws-subscriptions-clob.polymarket.com/ws/market to stream Level 1 (bids/asks). **Monitor:** Keep a local variable of the 'Best Ask' for the winning side. **The Trigger:** STRICTLY when Time Remaining <= 1 second (but > 0): Check if the winning side (Price > 0.50) is available below $0.99. **Safety:** Add a DRY_RUN boolean flag. If True, just print 'WOULD BUY'. If False, execute. **Action:** Send a 'Fill or Kill' (FOK) order for $0.99 using client.create_and_post_order. **Speed:** Use asyncio loop. Do not use time.sleep."

Part 4: How to Test Safely

Do not run this with full size immediately.

1. **Dry Run Mode:** In the generated sniper.py, ensure DRY_RUN = True. Run python sniper.py. Watch the terminal. If you see *"WOULD BUY YES at 0.98 [Time 10:14:59]"*, the logic is valid.
2. **The $1 Test:** Set DRY_RUN = False. Hardcode SIZE = 1 (1 share = ~$1). This limits your risk to exactly $1.00 while you test your internet latency.



Part 5: Execution

1. Open VS Code Terminal.
2. **Approve:** python approve.py (Only needed once).
3. **Scan:** python scanner.py (Get the IDs).
4. **Snipe:** Copy IDs into sniper.py and run python sniper.py.

Early Market Phase

Market launch

bet 1 - $0.01
bet 2 - $0.07
bet 3 - $0.12
bet 4 - $0.81

High upside
100x
potential

→ Low probability outcomes

Late Market Phase

Near Resolution

bet 1 - $0
bet 2 - $0.01
bet 3 - $0.02
bet 4 - $0.97

Low risk
1-3%
return

→ High probability outcome