

CS50's Introduction to Programming with Python

OpenCourseWare

Donate  (<https://cs50.harvard.edu/donate>)

David J. Malan (<https://cs.harvard.edu/malan/>)

malan@harvard.edu

 (<https://www.facebook.com/dmalan>)  (<https://github.com/dmalan>) 

(<https://www.instagram.com/davidjmalan/>)  (<https://www.linkedin.com/in/malan/>) 

(<https://www.reddit.com/user/davidjmalan>)  (<https://www.threads.net/@davidjmalan>)

 (<https://twitter.com/davidjmalan>)

Bitcoin Price Index

As of **Saturday, April 5, 2025 at 6:30 PM GMT+5:30**

(<https://time.cs50.io/20250405T130000Z>), we are aware of the **CoinCap v2 API deprecation** (<https://pro.coincap.io/api-docs/>) that may affect students' ability to complete this problem.

We have now updated the problem to use the CoinCap v3 API instead. If you have already started working on this problem, you must update your code to use the CoinCap v3 API instead of the CoinCap v2 API. Importantly, you will need to sign up for a CoinCap account and obtain an API key. You can do this by visiting **CoinCap** (<https://pro.coincap.io/signup>). Once you have your API key, you can use it in your code to access the CoinCap v3 API.

Bitcoin (<https://en.wikipedia.org/wiki/Bitcoin>) is a form of digital currency, otherwise known as **cryptocurrency** (<https://en.wikipedia.org/wiki/Cryptocurrency>). Rather than rely on a central authority like a bank, Bitcoin instead relies on a distributed network, otherwise known as a **blockchain** (<https://en.wikipedia.org/wiki/Blockchain>), to record transactions.

Because there's demand for Bitcoin (i.e., users want it), users are willing to buy it, as by exchanging one currency (e.g., USD) for Bitcoin.

In a file called `bitcoin.py`, implement a program that:



- Expects the user to specify as a command-line argument the number of Bitcoins, n , that they would like to buy. If that argument cannot be converted to a `float`, the program should exit via `sys.exit` with an error message.
- Queries the API for the CoinCap Bitcoin Price Index at rest.coincap.io/v3/assets/bitcoin?apiKey=YourApiKey (<https://rest.coincap.io/v3/assets/bitcoin?apiKey=YourApiKey>). You should replace `YourApiKey` with the actual API key you obtained from your CoinCap account dashboard, which returns a [JSON](https://en.wikipedia.org/wiki/JSON) (<https://en.wikipedia.org/wiki/JSON>) object, among whose nested keys is the current price of Bitcoin as a `float`. Be sure to catch any [exceptions](https://requests.readthedocs.io/en/latest/api/#exceptions) (<https://requests.readthedocs.io/en/latest/api/#exceptions>), as with code like:

```
import requests

try:
    ...
except requests.RequestException:
    ...
```

- Outputs the current cost of n Bitcoins in USD to four decimal places, using `,` as a thousands separator.

► Hints

Demo

This demo was recorded when the price of Bitcoin was \$38,761.0833. Your own output may vary.

```
$ python bitcoin.py
Missing command-line argument
$ python bitcoin.py cat
Command-line argument is not a number
$ python bitcoin.py 1
$38,761.0833
$ python bitcoin.py 1.5
$58,141.6249
$ python bitcoin.py 2
$77,522.1666
$
```

Recorded with [asciinema](#)

Before You Begin

Create a CoinCap account at [CoinCap Sign Up \(https://pro.coincap.io/signup\)](https://pro.coincap.io/signup) and obtain an API key by clicking the “**Add New Key**” button in your [CoinCap Dashboard \(https://pro.coincap.io/dashboard\)](https://pro.coincap.io/dashboard). You will need to use this API key in your program. You can read more about the API usage in the [CoinCap API documentation \(https://pro.coincap.io/api-docs/\)](https://pro.coincap.io/api-docs/).

Log into [cs50.dev \(https://cs50.dev/\)](https://cs50.dev/), click on your terminal window, and execute `cd` by itself. You should find that your terminal window’s prompt resembles the below:

```
$
```

Next execute

```
mkdir bitcoin
```

to make a folder called `bitcoin` in your codespace.

Then execute

```
cd bitcoin
```

to change directories into that folder. You should now see your terminal prompt as `bitcoin/ $`. You can now execute

```
code bitcoin.py
```

to make a file called `bitcoin.py` where you’ll write your program.

How to Test

Here’s how to test your code manually:

- Run your program with `python bitcoin.py`. Your program should use `sys.exit` to exit with an error message:

```
Missing command-line argument
```

- Run your program with `python bitcoin.py cat`. Your program should use `sys.exit` to exit with an error message:

```
Command-line argument is not a number
```

- Run your program with `python bitcoin.py 1`. Your program should output the price of a single Bitcoin to four decimal places, using `,` as a [thousands separator](#)

<https://docs.python.org/3/library/string.html#formatspec>).

- Run your program with `python bitcoin.py 2`. Your program should output the price of two Bitcoin to four decimal places, using `,` as a thousands separator (<https://docs.python.org/3/library/string.html#formatspec>).
- Run your program with `python bitcoin.py 2.5`. Your program should output the price of 2.5 Bitcoin to four decimal places, using `,` as a thousands separator (<https://docs.python.org/3/library/string.html#formatspec>).

You can execute the below to check your code using `check50`, a program that CS50 will use to test your code when you submit. But be sure to test it yourself as well!

```
check50 cs50/problems/2022/python/bitcoin
```

Green smilies mean your program has passed a test! Red frownies will indicate your program output something unexpected. Visit the URL that `check50` outputs to see the input `check50` handed to your program, what output it expected, and what output your program actually gave.

How to Submit

In your terminal, execute the below to submit your work.

```
submit50 cs50/problems/2022/python/bitcoin
```

