

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>)

Home Federal Savings Bank



In season 7, episode 24 (https://en.wikipedia.org/wiki/The_Invitations) of Seinfeld (<https://en.wikipedia.org/wiki/Seinfeld>), Kramer (https://en.wikipedia.org/wiki/Cosmo_Kramer) visits a bank that promises to give \$100 to anyone who isn't greeted with a "hello." Kramer is instead greeted with a "hey," which he insists isn't a "hello," and so he asks for \$100. The bank's manager proposes a compromise: "You got a greeting that starts with an 'h,' how does \$20 sound?" Kramer accepts.

In a file called `bank.py`, implement a program that prompts the user for a greeting. If the greeting starts with “hello”, output `$0`. If the greeting starts with an “h” (but not “hello”), output `$20`. Otherwise, output `$100`. Ignore any leading whitespace in the user’s greeting, and treat the user’s greeting case-insensitively.

▼ Hints

- Recall that a `str` comes with quite a few methods, per docs.python.org/3/library/stdtypes.html#string-methods (<https://docs.python.org/3/library/stdtypes.html#string-methods>).
- Be sure to give `$0` not only for “hello” but also “hello there”, “hello, Newman”, and the like.

Demo

```
Greeting: How you doing?
$20
$ python bank.py
Greeting: What's happening?
$100
$ python bank.py
Greeting: What's up?
$100
$ python bank.py
Greeting: How's it going?
$20
$
```

Before You Begin

Log into 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 bank
```

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

Then execute

```
cd bank
```

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

```
code bank.py
```

to make a file called `bank.py` where you'll write your program.

How to Test

Here's how to test your code manually:

- Run your program with `python bank.py`. Type `Hello` and press Enter. Your program should output:

```
$0
```

- Run your program with `python bank.py`. Type `Hello, Newman` and press Enter. Your program should output:

```
$0
```

- Run your program with `python bank.py`. Type `How you doing?` and press Enter. Your program should output

```
$20
```

- Run your program with `python bank.py`. Type `What's happening?` and press Enter. Your program should output

```
$100
```

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/bank
```

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/bank
```