

Isso é CS50

Introdução do CS50 à Ciência da Computação

OpenCourseWare

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

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

malan@harvard.edu

 (<https://www.clubhouse.com/@davidjmalan>)  (<https://www.facebook.com/dmalan>) 

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

(<https://www.linkedin.com/in/malan/>)  (<https://orcid.org/0000-0001-5338-2522>) 

(<https://www.quora.com/profile/David-J-Malan>)  (<https://www.reddit.com/user/davidjmalan>) 

(<https://www.tiktok.com/@davidjmalan>)  (<https://davidjmalan.t.me/>) 

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

Cash

Implement a program that calculates the minimum number of coins required to give a user change.

```
$ python cash.py
Change owed: 0.41
4
```

Getting Started

Log into code.cs50.io (<https://code.cs50.io/>), click on your terminal window, and execute `cd` by itself. You should find that your terminal window's prompt resembles the below:

```
$
```

Next execute

```
wget https://cdn.cs50.net/2022/fall/psets/6/sentimental-cash.zip
```

in order to download a ZIP called `sentimental-cash.zip` into your codespace.

Then execute

```
unzip sentimental-cash.zip
```

to create a folder called `sentimental-cash`. You no longer need the ZIP file, so you can execute

```
rm sentimental-cash.zip
```

and respond with “y” followed by Enter at the prompt to remove the ZIP file you downloaded.

Now type

```
cd sentimental-cash
```

followed by Enter to move yourself into (i.e., open) that directory. Your prompt should now resemble the below.

```
sentimental-cash/ $
```

Execute `ls` by itself, and you should see `cash.py`. If you run into any trouble, follow these same steps again and see if you can determine where you went wrong!

Specification

- Write, in a file called `cash.py`, a program that first asks the user how much change is owed and then spits out the minimum number of coins with which said change can be made. You can do this exactly as you did in [Problem Set 1](#), except that your program this time should be written in Python, and you should assume that the user will input their change in dollars (e.g., 0.50 dollars instead of 50 cents).
- Use `get_float` from the CS50 Library to get the user's input and `print` to output your answer. Assume that the only coins available are quarters (25¢), dimes (10¢), nickels (5¢), and pennies (1¢).
 - We ask that you use `get_float` so that you can handle dollars and cents, albeit sans dollar sign. In other words, if some customer is owed \$9.75 (as in the case where a newspaper costs 25¢ but the customer pays with a \$10 bill), assume that your program's input will be `9.75` and not `$9.75` or `975`. However, if some customer is owed \$9 exactly, assume that your program's input will be `9.00` or just `9` but, again, not `$9` or `900`. Of course, by nature of floating-point values, your program will likely work with inputs like `9.0` and `9.000` as well; you need not worry about checking whether the user's input is “formatted” like money should be.
- Se o usuário não fornecer um valor não negativo, seu programa deve solicitar novamente ao usuário um valor válido repetidas vezes até que o usuário cumpra.
- Aliás, para que possamos automatizar alguns testes do seu código, pedimos que a última linha de saída do seu programa seja apenas o número mínimo de moedas possível: um inteiro seguido de uma nova linha.

Uso

Seu programa deve se comportar de acordo com o exemplo abaixo.

```
$ python cash.py
Change owed: 0.41
4
```

teste

Embora `check50` esteja disponível para esse problema, você é encorajado a primeiro testar seu código por conta própria para cada um dos itens a seguir.

- Execute seu programa como `python cash.py`, e aguarde um prompt para entrada. Digite `0.41` e pressione enter. Seu programa deve produzir `4`.
- Execute seu programa como `python cash.py`, e aguarde um prompt para entrada. Digite `0.01` e pressione enter. Seu programa deve produzir `1`.
- Execute seu programa como `python cash.py`, e aguarde um prompt para entrada. Digite `0.15` e pressione enter. Seu programa deve produzir `2`.
- Execute seu programa como `python cash.py`, e aguarde um prompt para entrada. Digite `1.60` e pressione enter. Seu programa deve produzir `7`.
- Execute seu programa como `python cash.py`, e aguarde um prompt para entrada. Digite `23` e pressione enter. Seu programa deve produzir `92`.
- Execute seu programa como `python cash.py`, e aguarde um prompt para entrada. Digite `4.2` e pressione enter. Seu programa deve produzir `18`.
- Execute seu programa como `python cash.py`, e aguarde um prompt para entrada. Digite `-1` e pressione enter. Seu programa deve rejeitar essa entrada como inválida, solicitando novamente que o usuário digite outro número.
- Execute seu programa como `python cash.py`, e aguarde um prompt para entrada. Digite `foo` e pressione enter. Seu programa deve rejeitar essa entrada como inválida, solicitando novamente que o usuário digite outro número.
- Execute seu programa como `python cash.py`, e aguarde um prompt para entrada. Não digite nada e pressione enter. Seu programa deve rejeitar essa entrada como inválida, solicitando novamente que o usuário digite outro número.

Execute o abaixo para avaliar a exatidão do seu código usando `check50`. Mas certifique-se de compilar e testar você mesmo também!

```
check50 cs50/problems/2023/x/sentimental/cash
```

Execute o abaixo para avaliar o estilo do seu código usando `style50`.

```
style50 cash.py
```

Como enviar

Em seu terminal, execute o abaixo para enviar seu trabalho.

```
submit50 cs50/problems/2023/x/sentimental/cash
```

