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://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://twitter.com/davidjmalan)

Emojize

Because emoji aren’t quite as easy to type as text, at least on laptops and desktops, some programs support “codes,” whereby you can type, for instance, `:thumbs_up:`, which will be automatically converted to 👍. Some programs additionally support aliases, whereby you can more succinctly type, for instance, `:thumbsup:`, which will also be automatically converted to 👍.

See carpedm20.github.io/emoji/all.html?enableList=enable_list_alias (https://carpedm20.github.io/emoji/all.html?enableList=enable_list_alias) for a list of codes with aliases.

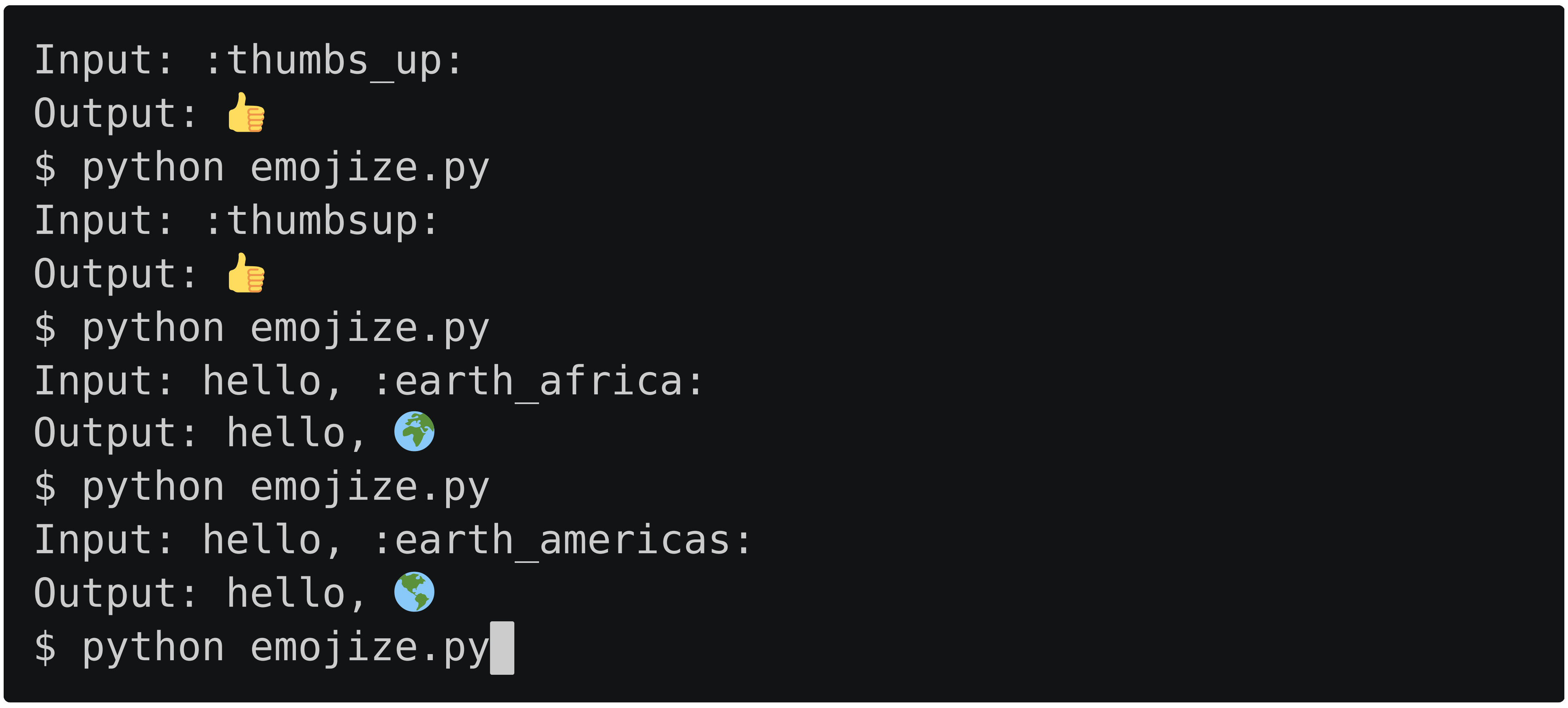
In a file called `emojize.py`, implement a program that prompts the user for a `str` in English and then outputs the “emojized” version of that `str`, converting any codes (or aliases) therein to their corresponding emoji.

▾ **Hints**

- Note that the `emoji` module comes with two functions, per pypi.org/project/emoji (https://pypi.org/project/emoji/), one of which is `emojize`, which takes an optional, named parameter called `language`. You can install it with:

  ```
  pip install emoji
  ```

Demo



```
Input: :thumbs_up:
Output: 👍
$ python emojize.py
Input: :thumbsup:
Output: 👍
$ python emojize.py
Input: hello, :earth_africa:
Output: hello, 🌍
$ python emojize.py
Input: hello, :earth_americas:
Output: hello, 🌎
$ python emojize.py
```

Recorded with asciinema

Before You Begin

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

```
mkdir emojize
```

to make a folder called `emojize` into your codespace.

Then execute

```
cd emojize
```

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

```
code emojize.py
```

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

How to Test

Here's how to test your code manually:

- Run your program with `python emojize.py`. Type `:1st_place_medal:` and press Enter. Your program should output:

  ```
  Output: 🥇
  ```

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

  ```
  Output: 💰
  ```

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

  ```
  Output: 😸
  ```

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/emojize
```

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/emojize
```