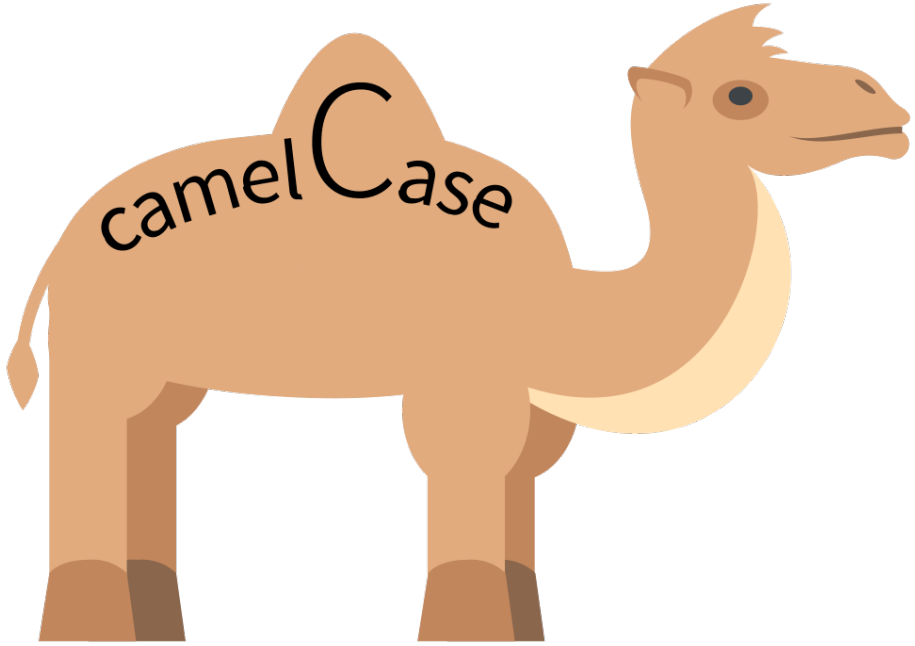
CS50's Introduction to Programming with Python

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)

camelCase



Source: en.wikipedia.org/wiki/Camel_case (https://en.wikipedia.org/wiki/Camel_case)

In some languages, it's common to use camel case (https://en.wikipedia.org/wiki/Camel_case) (otherwise known as "mixed case") for variables' names when those names comprise multiple words, whereby the first letter of the first word is lowercase but the first letter of each subsequent word is uppercase. For instance, whereas a variable for a user's name might be called `name`, a variable for a user's first name might be called `firstName`, and a variable for a user's preferred first name (e.g., nickname) might be called `preferredFirstName`.

Python, by contrast, recommends (https://peps.python.org/pep-0008/#function-and-variable-names) snake case (https://en.wikipedia.org/wiki/Snake_case), whereby words are instead separated by underscores (`_`), with all letters in lowercase. For instance, those same variables would be called `name`, `first_name`, and `preferred_first_name`, respectively, in Python.

In a file called `camel.py`, implement a program that prompts the user for the name of a variable in camel case and outputs the corresponding name in snake case. Assume that the user's input will indeed be in camel case.

▼ **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).
- Much like a `list`, a `str` is "iterable," which means you can iterate over each of its characters in a loop. For instance, if `s` is a `str`, you could print each of its characters, one at a time, with code like:

```
for c in s:
    print(c, end="")
```

Demo

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 camel
```

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

Then execute

```
cd camel
```

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

```
code camel.py
```

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

How to Test

Here's how to test your code manually:

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

```
name
```

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

```
first_name
```

- Run your program with `python camel.py`. Type `preferredFirstName` and press Enter. Your program should output

```
preferred_first_name
```

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/camel
```

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/camel
```