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)

Guessing Game

I'm thinking of a number between 1 and 100…

▶ **What is it?**

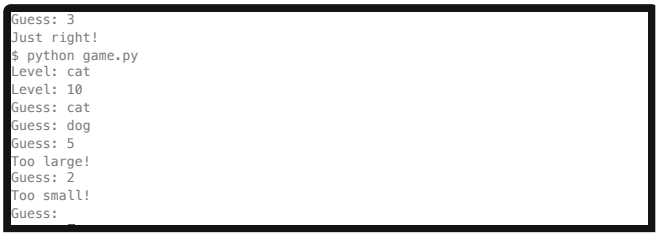
In a file called `game.py`, implement a program that:

- Prompts the user for a level, n. If the user does not input a positive integer, the program should prompt again.
- Randomly generates an integer between 1 and n, inclusive, using the `random` module.
- Prompts the user to guess that integer. If the guess is not a positive integer, the program should prompt the user again.
  - If the guess is smaller than that integer, the program should output `Too small!` and prompt the user again.
  - If the guess is larger than that integer, the program should output `Too large!` and prompt the user again.
  - If the guess is the same as that integer, the program should output `Just right!` and exit.

▼ **Hints**

- Note that the `random` module comes with quite a few functions, per docs.python.org/3/library/random.html (https://docs.python.org/3/library/random.html).

Demo



Recorded with asciinema

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 game
```

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

Then execute

```
cd game
```

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

```
code game.py
```

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

How to Test

Here's how to test your code manually:

- Run your program with `python game.py`. Type `cat` at a prompt that says `Level:` and press Enter. Your program should reprompt you:

  ```
  Level:
  ```

- Run your program with `python game.py`. Type `-1` at a prompt that says `Level:` and press Enter. Your program should reprompt you:

  ```
  Level:
  ```

- Run your program with `python game.py`. Type `10` at a prompt that says `Level:` and press Enter. Your program should now be ready to accept guesses:

  ```
  Guess:
  ```

- Run your program with `python game.py`. Type `10` at a prompt that says `Level:` and press Enter. Then type `cat`. Your program should reprompt you:

  ```
  Guess:
  ```

- Run your program with `python game.py`. Type `10` at a prompt that says `Level:` and press Enter. Then type `-1`. Your program should reprompt you:

  ```
  Guess:
  ```

- Run your program with `python game.py`. Type `1` at a prompt that says `Level:` and press Enter. Then type `1`. Your program should output:

  ```
  Just right!
  ```

  There's only one possible number the answer could be!

- Run your program with `python game.py`. Type `10` at a prompt that says `Level:` and press Enter. Then type `100`. Your program should output:

  ```
  Too large!
  ```

  Looks like you're guessing outside the range you specified.

- Run your program with `python game.py`. Type `10000` at a prompt that says `Level:` and press Enter. Then type `1`. Your program should output:

  ```
  Too small!
  ```

> Most likely, anyways: you might get lucky and see `Just right!`. But it would certainly be odd for you to see `Just right!` every time. And certainly you shouldn't see `Too large!`.

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/game
```

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/game
```