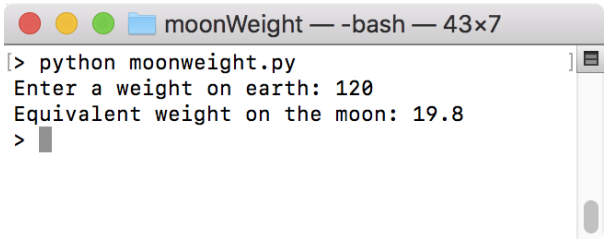
Chris Piech and Mehran Sahami
Code in Place

Section #2
April 22, 2020

Section Handout #2: Welcome to Python

Moonweight

Write a Python program that calculates a weight on the moon, given a weight on earth.



The user (an earthling) enters a weight on earth. Your program will print the corresponding weight on the moon, which is 16.5% of the weight on earth.

8-Ball

The idea behind an 8-Ball is very simple. You ask the eight ball a yes or no question, and it tells you the answer. Except, that the answer it choses is randomly selected from a set of prefabricated responses. Here is what a physical 8-ball looks like:

Write a program that continuously prompts the user for a yes or no question, and then **randomly** selects from a set of canned answers: The classic 8-ball responses were:

1. As I see it, yes.
2. Ask again later.
3. Better not tell you now.
4. Cannot predict now.
5. Concentrate and ask again.
6. Don't count on it.

7. It is certain.
8. It is decidedly so.
9. Most likely.
10. My reply is no.
11. My sources say no.
12. Outlook not so good.
13. Outlook good.
14. Reply hazy, try again.
15. Signs point to yes.
16. Very doubtful.
17. Without a doubt.
18. Yes.
19. Yes – definitely.
20. You may rely on it

You do not have to have so many responses. Instead choose at least 5, or make up your own. Here is a demo of the program running:

```
> python 8ball.py
Ask a yes or no question: Is Karel married?
Not a chance

Ask a yes or no question: Is my real name Chris?
Only Karel knows

Ask a yes or no question: 8-ball, are you using random numbers?
Without a doubt.
```

Milestone #1: For the basic version of the program, you only need to answer one question.

Milestone #2: If you have extra time, see if you can repeatedly answer questions until the user enters no question.

Useful Syntax

```
x = random.randint(A, B)
```

`randint` is a function that gives you back a random number in the range A to B inclusive. A and B could be literal numbers, constants, or variables. You can chose any name for the variable that stores the result.

```
if x == y:
    body
```

Is an if statement (just like in Karel!) but it passes under the condition that the value of *x* is the same as the value for *y*. *x* and *y* could be literal numbers, constants, or variables