Review for finals #1
Due Monday 5/17/21 by midnight

We are going to review Python for two weeks. Your final exam will be on Python. Today we're going to review variables, while loops, and if/elif/else structures.

First, log back in to repl. Create a new repl, name it "Review 1".

Type the following then run the program:

```
print("howdy")
```

Not too exciting, but we've got to ease back into everything so I figured we'd start super easy.

Remember how Python uses indentation to separate blocks of code? Each block of code has the same indent level. So in the following, the indented lines under the **while** line tell Python that those lines are inside the loop, while the line afterwards is not.

```
x = 0
while (x<4):
    print(x)
    x=x+1
print("done")
print("howdy!"*x)
```

The above prints out the following:

```
0
1
2
3
done
howdy!howdy!howdy!howdy!
```

Try it. Just reading through this review will not help you remember how to do things. Copy the above code, paste it after the howdy line in your current program, save and run it. You may have to fix the indenting after you paste it into your repl window. Change the code slightly, see what happens.

The last line in the above code prints a string (a piece of text) multiple times by using the * symbol with the variable x.

To put something into a variable, just write it with an equals sign: x = 0. To check if something is equal to something in an if statement you use two equals signs: if x == 0.

(continued on next page)

You can get data from a user using the **input** function:

```
   name = input("Please enter your name: ")

   or

   count = int(input("Please enter a number: "))
```

The second one above uses the **int()** function to change what the user entered into a number.

Saying "`print("hello"*count)`" will print "hello" **count** times on a single line.

You can loop using a while loop (see example on previous page, plus one here):

```
while True:
     n = input("Enter your favorite class: ")
     if n == "math":
          print("you win!")
          break
     else:
          print("try again")
          continue
```

If you want to break out of a loop early you can use the "**break**" command. If you want to go back to the top of a loop from inside an if statement you can use the "**continue**" command. In the above code you actually don't need the "continue" because you're in a while True: loop so it would already loop until you said "break". I just included the continue so you could see that there are multiple ways to do things.

You can evaluate variables using an if or an if/else statement (see example below):

```
while True:
     a = input("Please enter the best pet: ")
     if a == "dog":
          print("Great choice, best pet ever!")
          break
     elif a == "cat":
          print("Cats are pretty OK. Please try again")
     else:
          print("Please try for something a bit more traditional.")
```

That's a lot to remember. Let's try to put it to good use in a review assignment on the next page.

Today's task:

Empty out any practice work you did (the last two pages) from the repl, and then:

- Ask the user their name.
- Ask the user their age.
- Print your name as many times as the user's age.
- Use a while True: loop to ask the user to enter information. Don't make your program ask about pets or teachers, those are my examples.
- Use an if/elif/else statement to pass judgment on the user's entry. Make sure you have an if, an elif, and also an else in your code.
- Use a break to leave the loop when the user enters a particular answer, but after you comment on that answer.

Here's a sample run (yours must be different):

```
Please enter your name: Hays
Please enter your age: 55
HaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHa
ysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHays
HaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHaysHa
ysHaysHays
We're going to talk about teachers.
Please enter one of your favorite teachers: Sauron
Woah, that's a little dark. I'm going to ask you to try again.
Please enter one of your favorite teachers: Gollum
Meh, please try again.
Please enter one of your favorite teachers: Gandalf
A real good guy, great choice.
```

Remember that you can look at your old programs to see things you have done before. Jump in, have some fun, and get back in the Python swing.

Turn a share link in when you are done.