# Exercise 5: The dawn of the inputs

By: Rahul Sharma 18 April, 2018

Before going on with today's lecture, I'd like to tell that we're done with the printing exercises. Hurray! I know the previous exercises were not that challenging but they were important. If you haven't submitted your assignments for printing, please do them ASAP. Let's move on with the next concept: taking inputs from the user.

Taking an input from the user is a very important activity that a programming language like Python allows to do. Most of the times you'll be taking custom inputs from the users, calculating something using those inputs and finally give the desired output. Previous exercises have involved calculation of simple interest using variables which where the principal, rate and time. They were assigned some "hard" values inside the code. Taking inputs also allow us to change the inputs instead of changing the code every time you require a different result.

I'd like you to see how inputs are taken using the following exercise before we decide to elaborate on it later.

## Exercise 1: Understanding raw_input()

1. Create a file named **ex5_1.py** and write the following code in it:

```
print "How old are you?",
age = raw_input()
print "How tall are you?",
height = raw_input()
print "How much do you weigh?",
weight = raw_input()
print "So, you're %r old, %r tall and %r heavy." % (
age, height, weight)
```

2. Run the file, try to understand what is happening and write a comment above each line explaining that.

3. Go online and find out what Python’s raw_input does. Write the answer as a print statement. Be creative.

4. Can you find other ways to use it? Try some of the samples you find.

5. Write another “form” like the above code to ask some other questions. Write that in file **ex5_1_1.py**

6. Related to escape sequences, try to find out why the last line has '6\'2"' with that \' sequence. See how the single-quote needs to be escaped because otherwise it would end the string?

**BONUS: When I said "hard" values. What did you understand by it? Write it in a print statement.**

After doing the previous exercise, you noticed that we were asking questions from the user using the print statements. Well, the `raw_input()` has a property that it can ask the questions from the user itself without using print statements explicitly. You can put your questions inside the '(' and ')' (we call them parentheses characters) characters and `raw_input()` will convert the string inside the into a prompt.
An example can be: `age = raw_input("What is your age?")` which will save the input from the user in the `age` variable. This means that you can convert the code written in the previous exercise into something which has comparatively less number of line. The next exercise may be challenging because I'm not going to provide with any code. Interesting? Let's move on with the exercise.

**Exercise 2: Using raw_input( )**

1. Create a file named **ex5_2.py**

2. Rewrite the code written in **ex5_1.py** using the prompting abilities of the raw_input( ).

3. In Terminal, where you normally run python to run your scripts, type `pydoc raw_input`. Read what it says. If you're on Windows try `python - m pydoc raw_input` instead.

4. Get out of pydoc by typing q to quit.

5. Look online for what the pydoc command does.

6. Use pydoc to also read about open, file, os, and sys. It's alright if you do not understand those; just read through and take notes about interesting things.

*NOTE: The exercises in the coming chapters will become more and more challenging. Do not forget what you've already done. Keep on going through the lectures again and again and always ask questions whenever you're in doubt.*

*Next lecture: Taking arguments from the terminal.*