

---

---

# PYTHON ASSIGNMENT #1

---

---

In the first assignment, we are going to cover the following topics:

1. **Command line input and output**
2. **Variables and types**
3. **Doing some basic math**

You will have 2 classes to do this assignment. Good luck!

---

## DEMONSTRATION REVIEW

---

Elements of the Python language that we covered in class:

1. The **print** statement can be used to output a string of characters to the command line.

```
print "hello world"  
print "2 + 2 = ", 2+2  
print "The answer is", x
```

2. **Variables** are labels that point to stored data. Use the equal sign (=) to assign a value to a variable:

```
hydralisks = 42  
raynors_hitpoints = 99.03  
exit_message = "So long and thanks for all the fish"  
result = 7 < 2
```

Variables have different types, and mixing types can result in weird errors. The basic types of variables are:

- **Boolean:** This type of variable stores whether something is True or False (these are the only two values it can have).
- **Int:** This type of variable stores a number without any decimals – an integer. Example: 42
- **Float:** This type of variable stores a number with a decimal – a floating point number. Example: 3.14
- **String:** A string is a bunch of characters (letters, numbers, punctuation, etc.) that are “strung” together in a sequence, like a word or sentence. Example “Go go gadget copter!”

You can use the **type()** statement to determine the type of a value if you are ever in doubt. For example, `type(42)` will return `<type 'int'>` and `type("Moo!!!")` will return `<type 'string'>`

Because Python has **Dynamic Typing**, you can convert the type of a variable. Here are some examples:

`int(99.999)` will convert the float 99.999 into an int by chopping off the .9999

`float("3234.33")` will convert the string “3234.33” into the actual floating point number 3234.33



**EXERCISE 2. basicMath.py:** Write a program that takes 2 numbers from the user and does the basic math operations on them, like in the example screen shot here:

Again, Your output should look **EXACTLY** like the picture (it is part of the exercise to get it to match exactly)

**Worth 3 Marks:**

- 1 mark for proper comments at the top of your module.
- 1 mark for correctly accepting input from the user
- 1 marks for correctly calculating the math and outputting the answer exactly as in the example.

```
This program will do some simple math with two numbers
Give me a number 5
Give me another number 3

5.0 + 3.0 = 8.0
5.0 - 3.0 = 2.0
5.0 x 3.0 = 15.0
5.0 / 3.0 = 1.66666666667
5.0 % 3.0 = 2.0
5.0 ^ 3.0 = 125.0
>>> |
```

**EXERCISE 3: years2seconds.py** Write a program that converts years into seconds. The user should be prompted to enter an amount of time in years and the program will then return the time converted into seconds. Don't worry about leap years and all that; just go with the basics.

```
>>>
This program converts years into seconds

How many years old are you? 30

Wow, then you are 946080000.0 seconds old
>>>
```

Your output should look **EXACTLY** like the picture (it is part of the exercise to get it to match exactly)

**Worth 4 Marks:**

- 1 mark for proper comments at the top of your module.
- 1 mark for correctly accepting input from the user.
- 1 mark for correctly calculating the conversion.
- 1 mark for outputting the answer exactly as in the example.

**EXERCISE 4. madlib.py:** Write a “Mad Lib” program as demonstrated in class.

Sample Output:

```
IDLE 1.2.1      ==== No Subprocess ====
>>>
This program simulates a Mad Lib.  Enjoy! ← #1

Noun: toaster
Plural Noun: robots
Noun: gravy
Plural Noun: starbucks coffee filters } #2
Adjective: wonderful
Noun: radio
Plural Noun: DVDs
Noun: orange

===== MAD LIB =====

A Ghost is supposed to be the spirit of a dead toaster. ← #4
Most ghosts appear as white robots floating in the gravy.
Ghosts are known to haunt starbucks coffee filters.

A Goblin is believed to be a wonderful creature with a misshapen radio.
Like all wicked DVDs, they delight in scaring the orange out of you.

Press enter to exit| ← #6
```

**Worth 8 Marks:**

- 1 mark for proper comments at the top of your module.
- 2 marks for correctly accepting all 8 inputs from the user
- 4 marks for formatting the output exactly like in the sample question. (see the following “Formatting Concerns” to know what to watch out for
- 1 mark for writing your own interesting mad lib.

### Formatting Concerns

**#1:** You should have a header that explains the program

**#2:** You should have 8 input prompts on 8 separate lines. There should be a space between the prompt and the place where you type in the value.

**#3:** Your work should be formatted nicely as in the sample output. Notice that punctuation like periods come right after the user's inputted values (see **#4**). Watch out that you don't have any weird spaces (like **#5**). You can avoid weird spaces by understanding the difference between using “,” and “+” in print statements.

**#6** you can ignore, you don't need it right now (ask me and I'll tell you why)