

# Assignment 6

---

## Python Operators and User Input

### INSTRUCTIONS

For all exercises, create Python scripts as described in the exercise instructions. Save these scripts in separate files named A6En.py, where n is the exercise number. Submit all script files to Gradescope Assignment 6. For each script, be sure following the best practices discussed in class, including:

- Descriptive variable names
- Consistent variable naming convention (snake\_case or camelCase)
- Module docstring that describes the script functionality and how to run it from the command line
- Block and/or inline comments describing portions of code that are not obvious

Upon submission, your scripts will automatically be graded for functional correctness. Gradescope will generate a report that indicates the pass/fail result of each automated test. For failed tests, the report will describe why it failed and suggest possible sources of error in your script.

You may correct errors and resubmit your scripts as many times as you like before the due date.

### EXERCISE 1

Write a Python script that accepts one command line parameter that is a person's name, and then prints the following three complimentary sentences. Script output must be exactly as shown in the example script output below.

Example script output:

```
PS C:\> python A6E1.py Jeremy
```

```
Jeremy is the most amazing person I have ever met.  
My mother is always saying, "You need to be more like Jeremy."  
Dear Jeremy: You rock!
```

```
PS C:\> python A6E1.py 'Professor Melo'
```

```
Professor Melo is the most amazing person I have ever met.  
My mother is always saying, "You need to be more like Professor Melo."  
Dear Professor Melo: You rock!
```

Hints:

- To use command line parameters, the script must import argv from the sys module (Check the Module 6 PDF for a refresher)
- The first command line parameter value is in `argv[1]`.

## GRADESCOPE SUBMISSION

A Python script named `A6E1.py`

## EXERCISE 2

Write a Python script that accepts one command line parameter that is a temperature in degrees Fahrenheit (F), and then prints out the equivalent temperatures in degrees Celsius (C) and Kelvin (K). All three temperatures printed by the script must be rounded to 2 decimal places. Script output must be exactly as shown in the example script output below.

Example script output:

```
PS C:\> python A6E2.py 32
```

```
32.00 F == 0.00 C == 273.15 K
```

```
PS C:\> python A6E2.py 0
```

```
0.00 F == -17.78 C == 255.37 K
```

```
PS C:\> python A6E2.py -459.67
```

```
-459.67 F == -273.15 C == 0.00 K
```

```
PS C:\> python A6E2.py 85.00234
```

```
85.00 F == 29.45 C == 302.60 K
```

Hints:

- Command line parameter values in `argv` are always strings and must be converted to numbers to be used in an arithmetic expression.
- Use the temperature conversion formulas from here: <https://www.nist.gov/pml/owm/si-units-temperature>
- Use parentheses to force the correct order of arithmetic operator evaluation.
- F-string format specifiers must be used to round printed numbers as `print(f"Only show {variable:.2f} decimal places")`
- Do not round the degrees Fahrenheit value before calculating the other temperatures, since it would cause an unnecessary loss of precision. It is best practice to round only when displaying numbers.

## GRADESCOPE SUBMISSION

A Python script named `A6E2.py`

## EXERCISE 3

Write a Python script that asks the user to enter a distance in inches, and then prints out the equivalent distance in feet and inches. If the user enters a value that is not an integer, the script must print a descriptive error message. Script output must be exactly as shown in the example script output below.

Example script output:

PS C:\> python A6E3.py

```
Enter a distance in inches: 123
123" == 10'3"
```

PS C:\> python A6E3.py

```
Enter a distance in inches: -345
-345" == -29'3"
```

PS C:\> python A6E3.py

```
Enter a distance in inches: paneer
Error: Distance in inches must be an integer.
```

PS C:\> python A6E3.py

```
Enter a distance in inches: 12.34
Error: Distance in inches must be an integer.
```

Hints:

- ' is the symbol for feet and " is the symbol for inches. There are 12" in 1'.
- The `input()` function always returns a string that must be converted to a number to be used in an arithmetic expression.
- The arithmetic expressions for this script will be like the example on lecture slide 17.
- Don't use magic numbers! Create a named constant like `INCHES_PER_FEET = 12` and use it in the calculations.
- The `int()` function will throw an exception if its argument cannot be converted to an integer. Use `try` and `except` to catch such an exception and print a descriptive error message if the value entered by the user is not a valid integer. See the examples on lecture slides 45 and 57.

## DOCSTRING SAMPLE

This is the style that the auto-grader is looking for. By now you're aware of paying attention to where newlines are and formatting conditions! Details are everything!

```
"""
This script accepts a name as a command-line argument and prints out a
personalized message.

Example usage:
python A6E1.py Jeremy

Author: Abe the Sorcerer
"""
```

## GRADESCOPE SUBMISSION

A Python script named **A6E3.py**