

Unit 3: Text-Based Programming

Lesson 6

Variable Naming

When making variables you should follow some general guidelines:

- Use a short but descriptive name
- You can not command words as functions (ex. print can not be a variable)
 - <https://docs.python.org/3/library/functions.html> -> link to built in functions, there are also built in items
- Spaces are not allowed
- Can't start with a number
- Use lower case separated with underscores (ex. first_name) or use CamelCase (ex. FirstName)

Commenting

You should use commenting to explain what your code is doing.

A `#` is used for commenting, you may have noticed when you open a new file in Mu that there is a comment in the blank file.

The IDE will not process anything written after a `#` on a line, so you can put your comment on the same line as a command.

```
inch_height = input("What is your height in inches? ") # Ask the user for their height
```

Commenting

Commenting is important, it will be used by others that view your code to explain what you are doing and it is also useful for yourself when you go back to a program that you created in the past to help you remember what you did.

If you are going to write multiple lines of comments then you should use docstrings(example on next page)

You can use commenting or docstrings to give a description of your program at the beginning of the file, before you write any lines of code.

Using # for comments

```
# This program is used to add two numbers

num_1 = int(input("First number: ")) #get the first number from user
num_2 = int(input("Second number: ")) #get the second number from user

result = num_1 + num_2 #get the sum of both numbers

print(f'{num_1} + {num_2} = {result}') #display the result
```

Using docstrings at the start of a program.

```
'''
This is a doc string.
It can be used to write multi-line comments and you
don't have to put a # at the start of each line, but you
do need to put three apostrophes at the end.

This program is used to add two numbers.
'''

num_1 = int(input("First number: "))    #get the first number from user
num_2 = int(input("Second number: "))   #get the second number from user

result = num_1 + num_2    #get the sum of both numbers

print(f'{num_1} + {num_2} = {result}')  #display the result
```

Commenting

From now on you should add comments to all of your projects that you create.

On the first assignment you will be asked to put a comment on every line of code to explain what it does. But after that you can explain sections of your code.

```
# This program is used to add two numbers

# get the numbers from user and add them
num_1 = int(input("First number: "))
num_2 = int(input("Second number: "))
result = num_1 + num_2

print(f'{num_1} + {num_2} = {result}') #display the result
```

Best Practices

- Planning is necessary so you have a clear vision of what you are creating and the general process for building the blocks of code.
 - Flowchart
 - Pseudocode
- Optimizing code is essential so that it is efficient and requires as few lines as possible (the tutorials are usually longer than they need to be).

- Documenting should be practiced within the code to provide details on what the code is doing. Good documentation is essential in all programming projects.
- Simplifying code is very important. Keeping code straightforward means there should be less bugs and it will take less time to debug a piece of code when testing.

- Collaboration between programmers is a common. You are encouraged to review, comment on and debug each others code. Getting someone else to test your program can often expose errors that you may not have thought of.
 - This does not mean you share your assignment with another student so that they don't have to do any work.