* **DEFINITIONS**
* The area where we’ve been writing our code is called the *editor*.
  + # is the token for single line comments.
* The *console* is where the results of your code is shown.
* **COMMENTS**
* # This is a sample comment!
  + # is the token for single line comments.
* ‘’’ This is a multi-line comment,

as you may see here. ‘’’

* A set of triple quotation marks will enable multi-line comments.
* **INPUT**
* name = input(“What is your name? “)
  + Get input from the user.
* **OUTPUT**
* print(“Welcome to Python!”)
  + print is a keyword.
  + No semi-colons are needed.
* print(my\_variable)
* print(my\_name)
* print(“Life “ + “of “ + “Brian”)
  + Combining strings together like this is called *concatenation*.
* string\_1 = “Camelot”

string\_2 = “place”

print(“Let’s not go to %s. ‘Tis a silly %s.” % (string\_1, string\_2))

* + This is an example of string formatting with the % operator.
* print(“Let’s not go to {}. ‘Tis a silly {}.”.format(string\_1, string\_2))
* **VARIABLES**
* my\_variable = 10
  + No variable keyword is needed.
  + We can also declare floats and booleans similarly.
* my\_name = “Kevin\’s Name”
  + This is how we can can assign a value to a string variable.
  + We use a backslash before the apostrophe to indicate that the apostrophe is a character in the string.
* **BOOLEANS**
* my\_bool = True
  + No keyword is needed. my\_bool is simply a variable name.
  + Capitalize the first letter of the boolean value.
* **MATH**
* addition = 72 + 23
  + addition is a variable name for a variable that will store the sum of 72 and 23.
* exponentiation = 10 \*\* 2
  + exponentiation is equal to 100.
* modulo = exponentiation % 12
  + modulo is equal to 4 since 100 mod 12 = 4.
* **ARRAYS**
* fifth\_letter = “MONTY”[4]
  + The variable fifth\_letter is assigned to the fifth letter of the string “MONTY” which is accessed by the fourth index.
* **METHODS**
* Methods that use dot notation only work with strings. Other methods can work on other data types.
* print len(my\_string)
  + This will print the number of characters in the string. The null character is not included.
* parrot = “Norwegian Blue”

print parrot.lower()

* Lowercase all letters in the string.
* print parrot.upper()
  + Uppercase all letters in the string.
* print “The value of pi is around “ + str(3.14)
  + Convert the double data type to a string data type.
* **The datetime Library**
* from datetime import datetime

now = datetime.now()

print now

* + 2017-10-02 02:30:48.623105
* current\_year = now.year

current\_month = now.month

current\_day = now.day

* + If we printed these variables, these integers would be our output:
    - 2017

10

2

* now.hour, now.minute, now.second
* **Control Flow**
* Gives us the ability to choose among outcomes based off what else is happening in the program.
* Comparators
  + ==, !=, <, <=, >, >=
* Boolean operators
  + not, and, or (operator precedence)
* Conditional statements
  + if 8 < 9:

print “You’re sane!”

elif some\_variable > 11:

print “What happened to ‘else if’?”

else:

print “Remember the colon!”

* **FUNCTIONS**
* def using\_control\_one():

# some code here, but we don’t need braces later on to close the function

# we just need good indentation to signal the end of a function

* + print is a keyword.
  + No semi-colons are needed.
* **VARIABLES**
* my\_variable = 10
  + No variable keyword is needed.
* **OUTPUT**
* print “Welcome to Python!”
  + print is a keyword.
  + No semi-colons are needed.
* **VARIABLES**
* my\_variable = 10
  + No variable keyword is needed.
* **OUTPUT**
* print “Welcome to Python!”
  + print is a keyword.
  + No semi-colons are needed.
* **VARIABLES**
* my\_variable = 10
  + No variable keyword is needed.
* **OUTPUT**
* print “Welcome to Python!”
  + print is a keyword.
  + No semi-colons are needed.
* **VARIABLES**
* my\_variable = 10
  + No variable keyword is needed.