# Types of Programming Errors

1. **Syntax Errors**
   - Errors in the programming language "grammar".
   - Python identifies these errors by highlighting the line the error is located on and giving the programmer a message describing the error.
   - These errors must be corrected before the program will successfully execute.
   - Examples include:
     1) print "Hello" *# No brackets*
     2) Print (hello) *# Capital, no quotes*
     3) for i in range (1,10) *# No colon (:)*
     4) Python indentation errors

2. **Run-Time Errors**
   - Errors that occur during the execution of the program.
   - They are often caused by incorrect user input which has not been anticipated by the programmer.
   - The programmer can often "recover from these errors gracefully"
   - Examples include:
     1) avg = 7 / *num* *# A run time error if the user has inputted 0 for num*
     2) Not including an *else* in a menu selection structure to act as a "catch all" if the user .

3. **Logic Errors**
   - Errors in the logic of the algorithm (ie. the solution).
   - The computer is a dumb machine and will do what you tell it (GIGO)!
   - These errors can only be identified by thoroughly testing the program as you are coding.
   - Examples include:
     1) for i in range (1,10)*: # Output the numbers between 0 and 10*
        print (i)
     2) square = num * 2 *# Stores the square of the number*

4. **Documentation Errors**
   - Errors in the internal documentation of the program (ie. program description not done, functions, parameters and/or variables not described, etc.)
   - Also include creating a program which is not user-friendly (ie. lack of on-screen instructions for the user, failure to identify output, etc.)
   - These errors can be identified by having others test your program and examine your documentation.