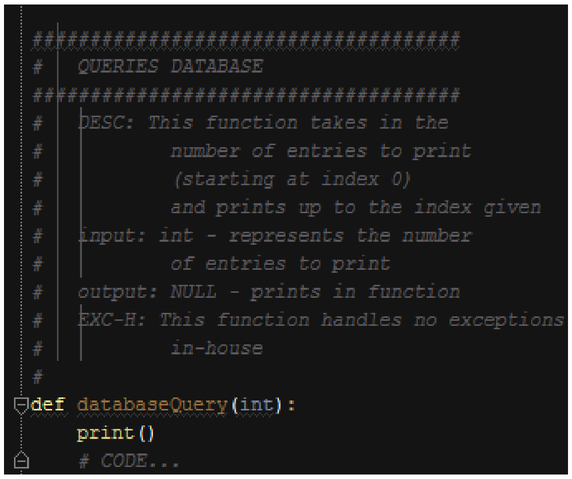
Coding Standards

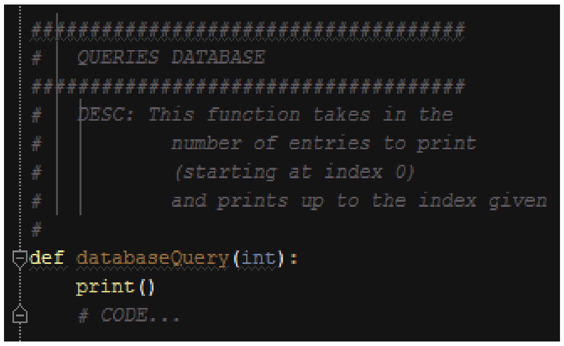
REVISION DATE	REASON	NAME(S)
3/10/16	*Doc. Created*	*Aaron Starr*

Comment Blocks:

Begin with line of comment variables, descriptive title of function or structure, comment variables, function variables



```
######################################
#   QUERIES DATABASE
######################################
#   DESC: This function takes in the
#           number of entries to print
#           (starting at index 0)
#           and prints up to the index given
#   input: int - represents the number
#           of entries to print
#   output: NULL - prints in function
#   EXC-H: This function handles no exceptions
#           in-house
#
def databaseQuery(int):
    print()
    # CODE...
```



```
######################################
#   QUERIES DATABASE
######################################
#   DESC: This function takes in the
#           number of entries to print
#           (starting at index 0)
#           and prints up to the index given
#
def databaseQuery(int):
    print()
    # CODE...
```

LEFT: A more formal comment block with good information, more can be added if desired.
RIGHT: Bare minimal, at least say what is happening inside the function

Camel-Casing:

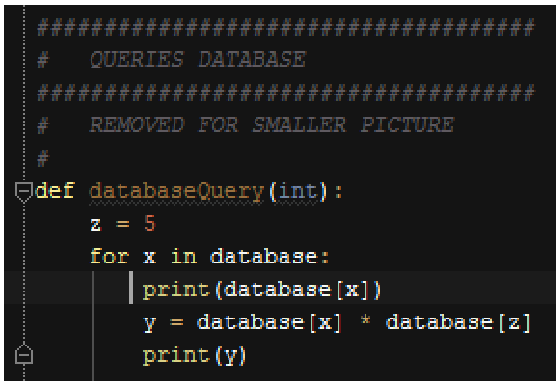
Camel-casing is the practice of creating a variable that is multiple words and distinguishing between words using capital letters in the new words. For example:

"databaseQuery" ::: starts with the first word lowercase and the next has a capital start
"databaseQueryListOfResidents" ::: this example shows a larger string of words using camel-casing

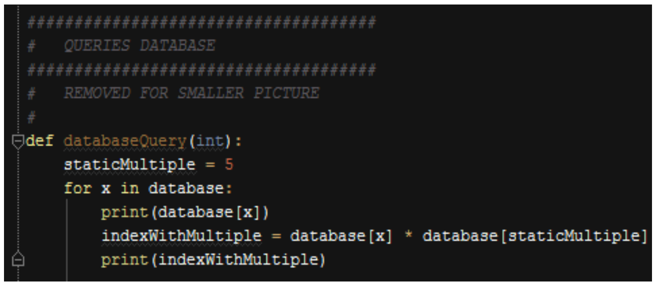
NOTICE: The first word is lowercase, this is the typical convention of camel-casing

Naming Conventions:

Naming a variable, function, class, or else is very important when there is a lot of code. Thus name variables wisely.



```
########################################
#   QUERIES DATABASE
########################################
#   REMOVED FOR SMALLER PICTURE
#
def databaseQuery(int):
    z = 5
    for x in database:
        print(database[x])
        y = database[x] * database[z]
        print(y)
```

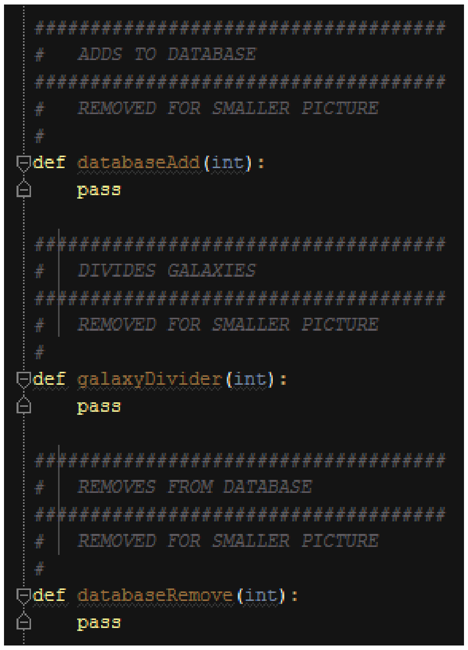


```
####################################
#   QUERIES DATABASE
####################################
#   REMOVED FOR SMALLER PICTURE
#
def databaseQuery(int):
    staticMultiple = 5
    for x in database:
        print(database[x])
        indexWithMultiple = database[x] * database[staticMultiple]
        print(indexWithMultiple)
```

LEFT: Not a lot of information given by the variables. Very generic names
RIGHT: Names that have some meaning. Notice x did not change, variables that remain are used for iteration are typically left bland, because they have little involvement.

Functional Organization:

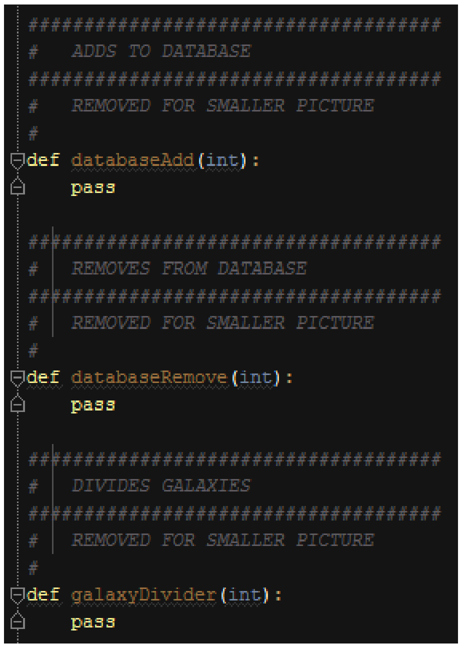
In this particular project, most code is placed into a flat file, a file that contains a large number of functions. To combat the stress of this we can organize the functions, classes, etc. together in terms of functionality or whatever is decided.



```
########################################
#   ADDS TO DATABASE
########################################
#   REMOVED FOR SMALLER PICTURE
#
def databaseAdd(int):
    pass

########################################
#   DIVIDES GALAXIES
########################################
#   REMOVED FOR SMALLER PICTURE
#
def galaxyDivider(int):
    pass

########################################
#   REMOVES FROM DATABASE
########################################
#   REMOVED FOR SMALLER PICTURE
#
def databaseRemove(int):
    pass
```



```
########################################
#   ADDS TO DATABASE
########################################
#   REMOVED FOR SMALLER PICTURE
#
def databaseAdd(int):
    pass

########################################
#   REMOVES FROM DATABASE
########################################
#   REMOVED FOR SMALLER PICTURE
#
def databaseRemove(int):
    pass

########################################
#   DIVIDES GALAXIES
########################################
#   REMOVED FOR SMALLER PICTURE
#
def galaxyDivider(int):
    pass
```

LEFT: Not organized as well as it could be.
RIGHT: A reasonable organizational scheme