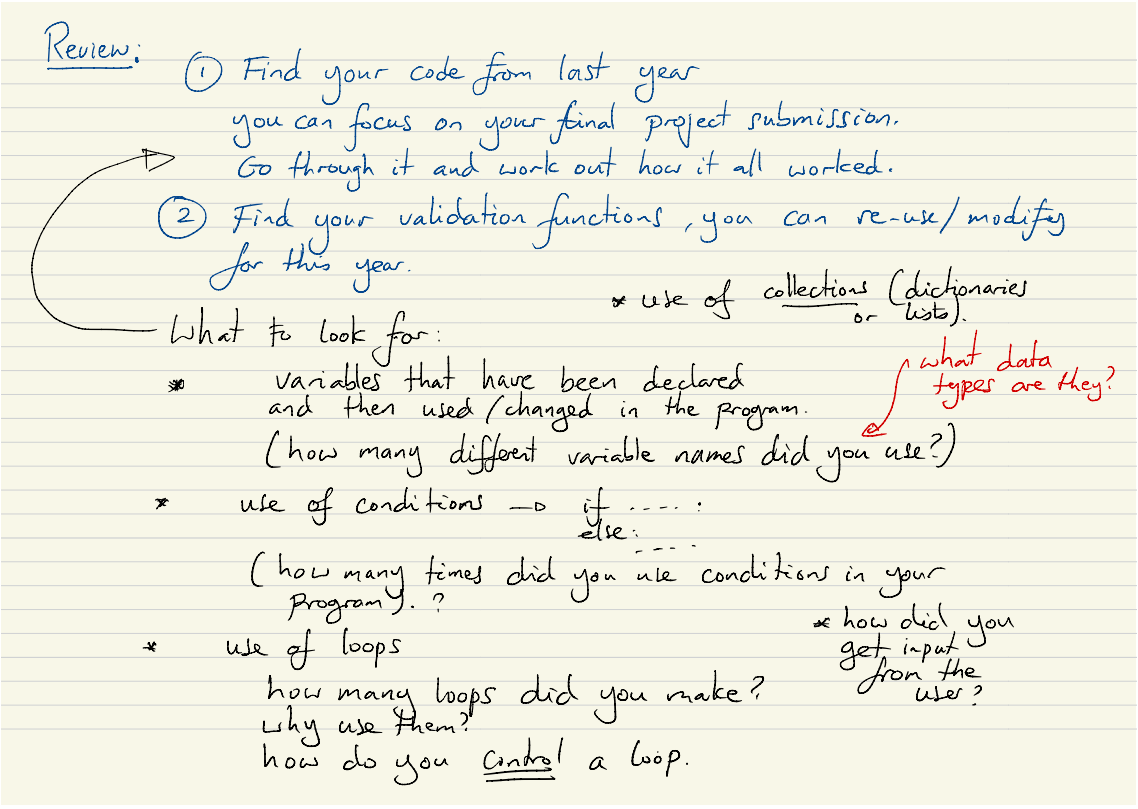
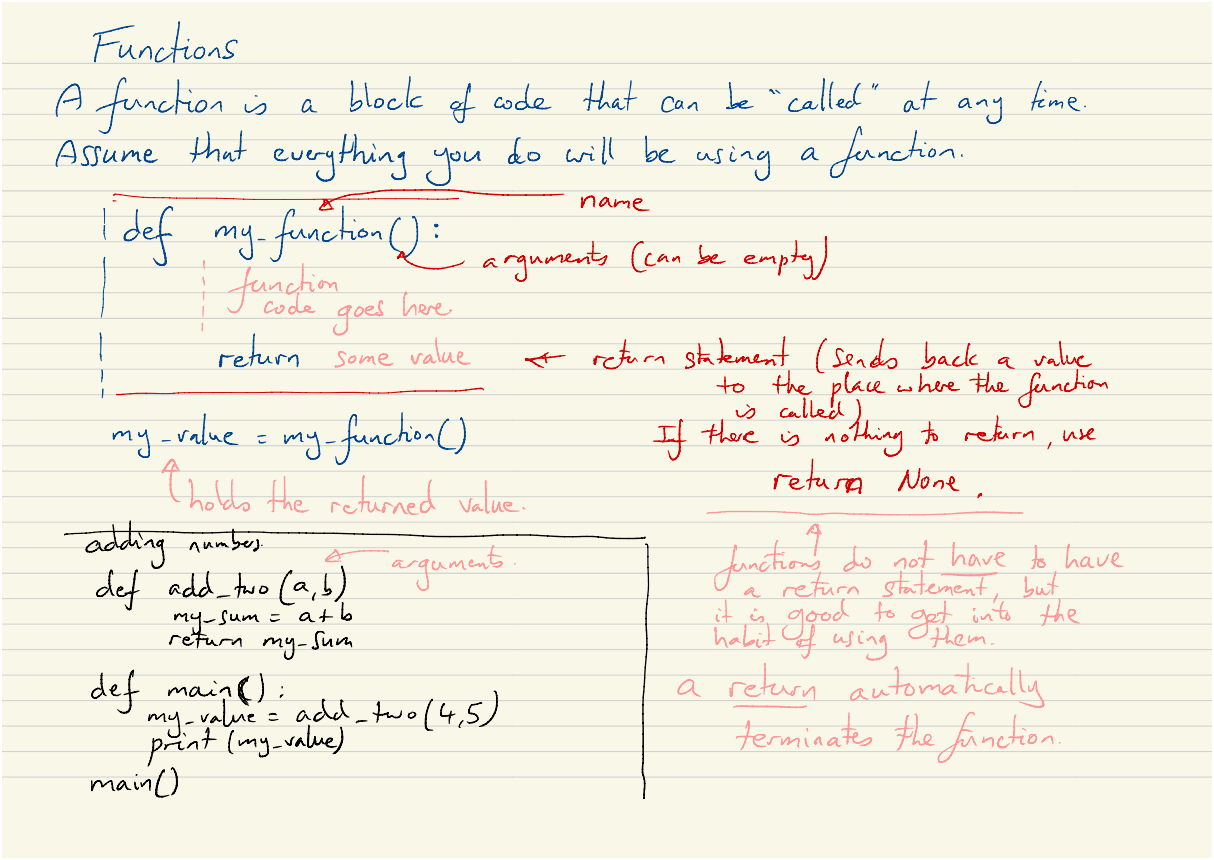
1 Review


Review:
① Find your code from last year
you can focus on your final project submission.
Go through it and work out how it all worked.
② Find your validation functions, you can re-use/modify for this year.
What to look for:
* use of collections (dictionaries or lists).
* variables that have been declared and then used/changed in the program.
(how many different variable names did you use?)
what data types are they?
* use of conditions → if ... :
else: ...
(how many times did you use conditions in your program?)
* use of loops
how many loops did you make?
why use them?
how do you control a loop.
* how did you get input from the user?

2 Functions

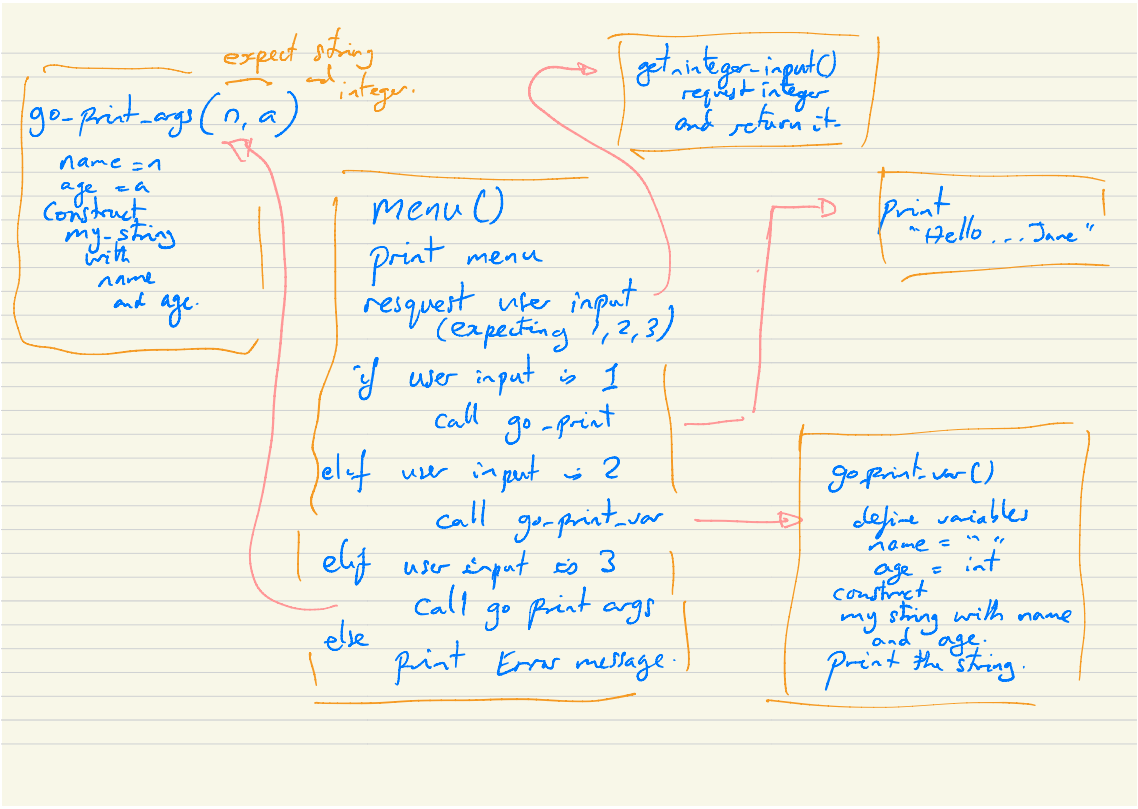


- Functions allow programmers to avoid creating code that has lots of repetitions.
- Functions allow programmers to break the structure of a program into blocks (or modules), which means that different problems can be separated out and dealt with independently.

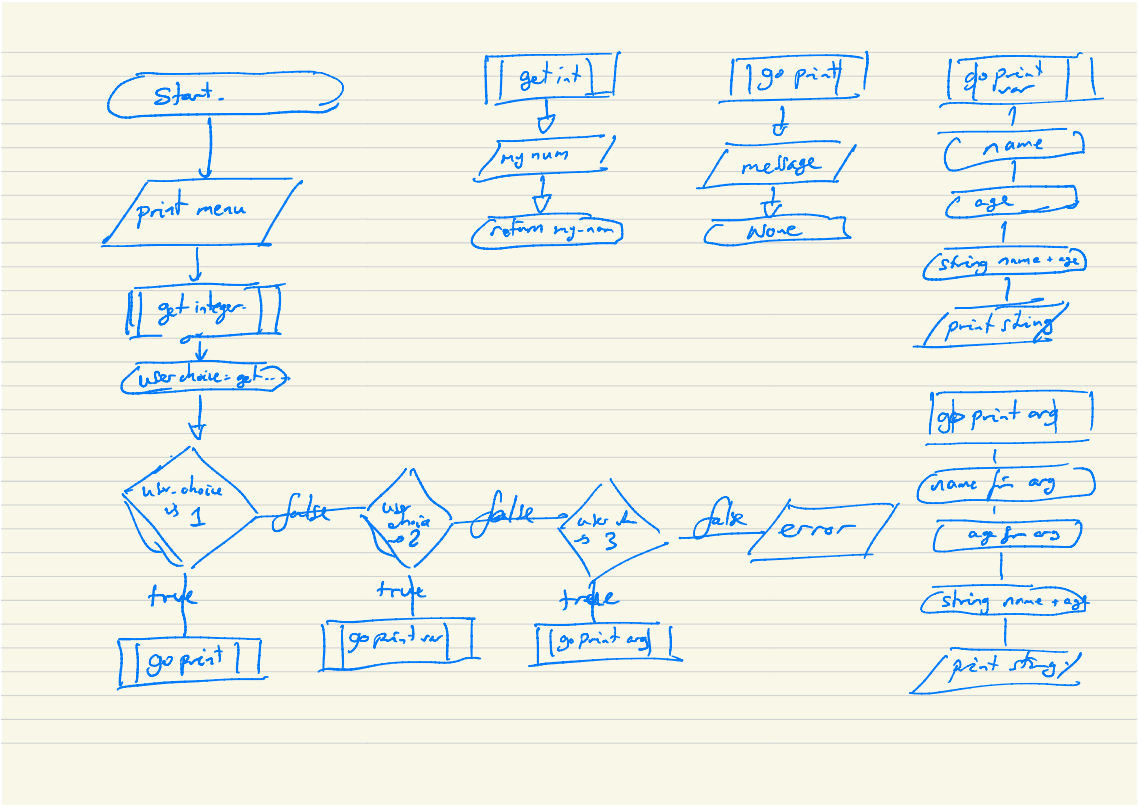
```
1
2
3  def go_print():
4      print("Hello my name is Jane")
5
6  def go_print_var():
7      name = "Bob"
8      age = 10
9      # put them together in a string
10     my_string = "My name is {} \nI am {} years old".format(name, age)
11     print(my_string)
12
13 def go_print_args(n,a):
14     name = n
15     age = a
16     my_string = "My name is {} \nI am {} years old".format(name, age)
17     print(my_string)
```

```
18
19 def get_integer_input():
20      my_number = int(input("please enter number: -> "))
21      return my_number
22
23 def menu():
24      run = True
25      while run is True:
26           my_menu = '''
27                     1: go print
28                     2: go print var
29                     3: go print args
30                     0: quit
31                     '''
32           print(my_menu)
33           choice = get_integer_input()
34           if choice == 1:
35                go_print()
36           elif choice == 2:
37                go_print_var()
38           elif choice == 3:
39                go_print_args("Jill", 0)
40           elif choice == 0:
41                run = False
42                print("Thank you")
43           else:
44                print("you have not chose a valid option")
45
46 menu()
```

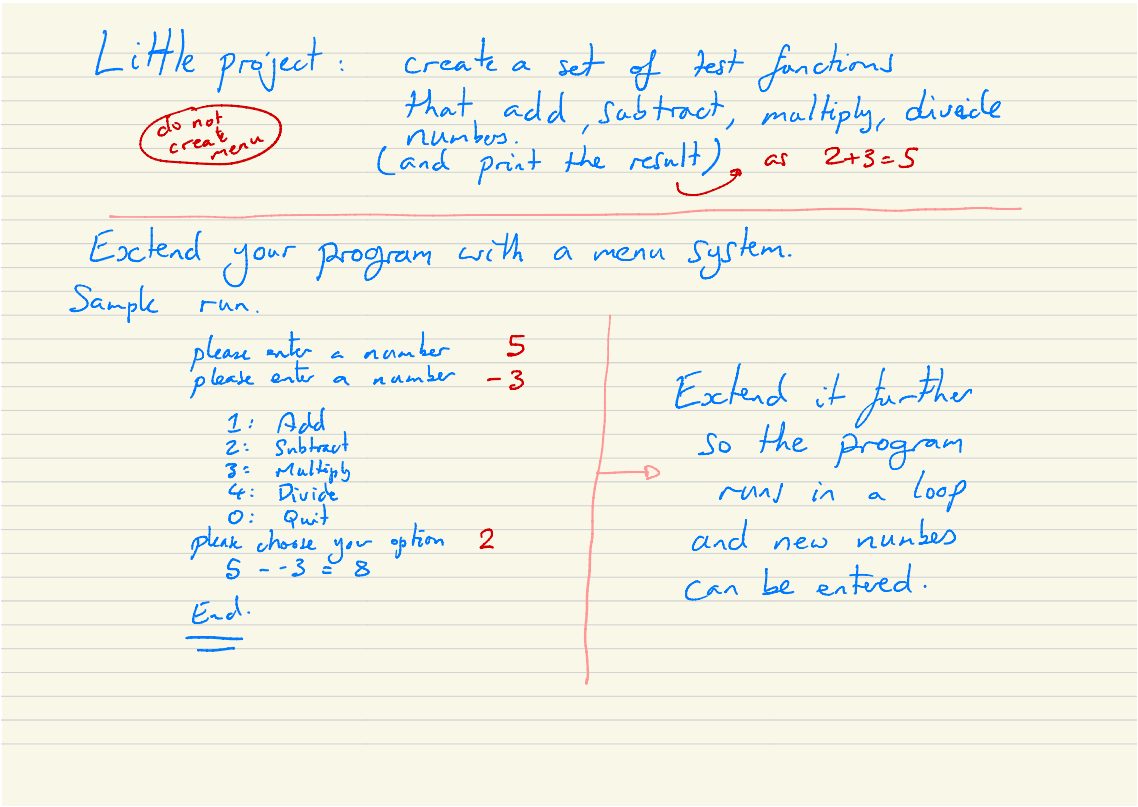
Listing 1: Test program - functions and menu



Simple Plan



Simple Plan as flow diagram



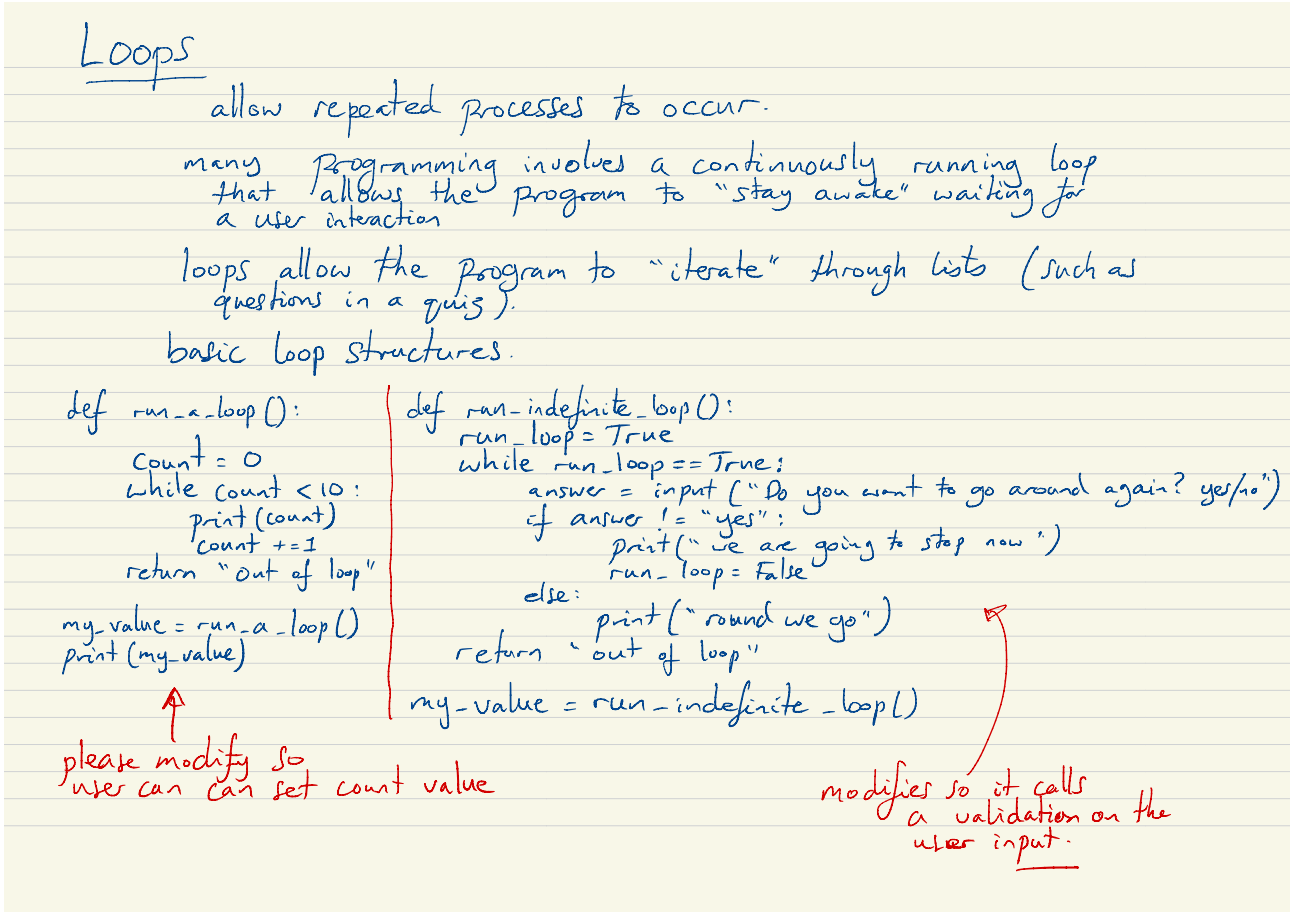
Little start project

3 String Formatting

Manging strings for print output is critical:

Please review using:
https://www.w3schools.com/python/python_string_formatting.asp
This means that you actively make variations on the examples given on the page.

4 Loops



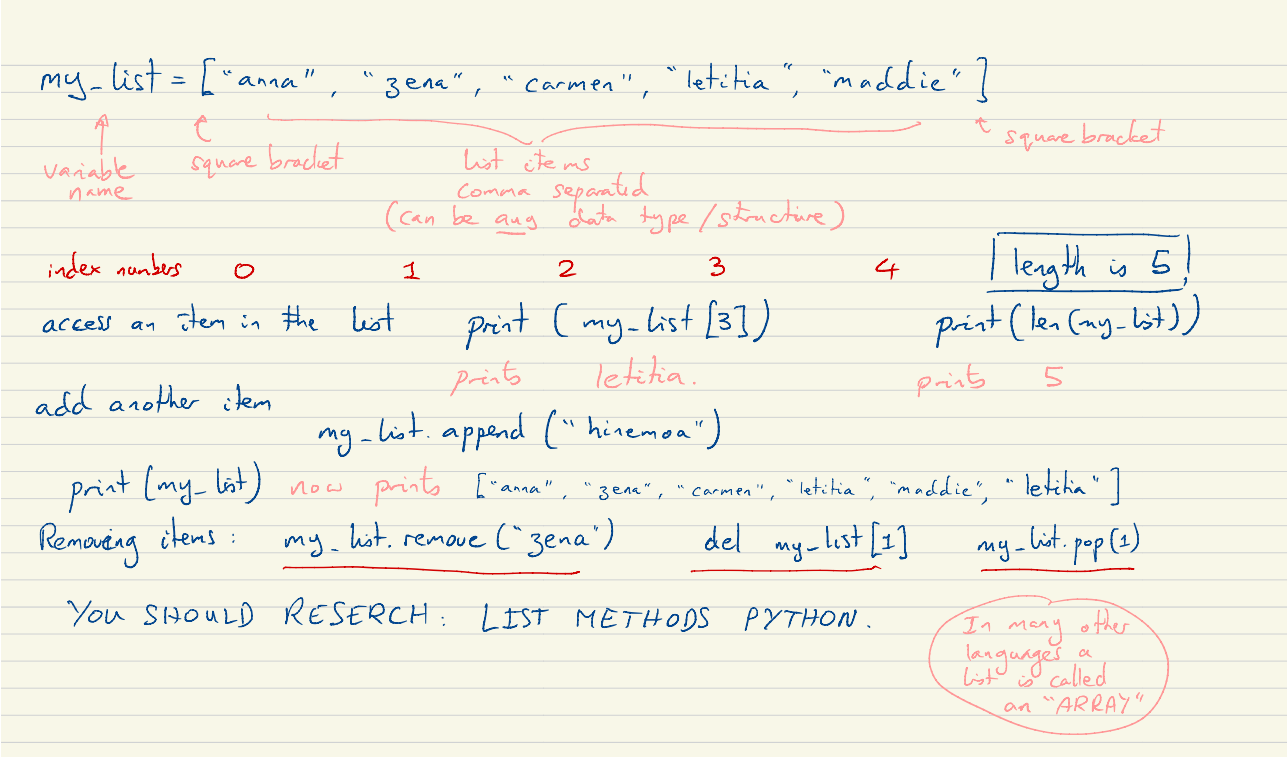

```
def while_loop_counter(count):
    # requires a counter
    c = 0
    # condition is set on the counter value
    while c < count:
        print("hello")
        # counter must be incremented
        c +=1
    print("All done")


def while_loop_indefinite():
    # set a condition
    run = True
    # check the condition
    while run is True:
        choice = input("Enter Q to quit or any other key to stay where you are: -> ")
        if choice == "Q":
            print("See you later")
            run = False
```

```
        else:
            print("I am still here")


def for_in_range_loop():
    # this has a built in counter
    # can be anything but we generally use i or j or k
    # can add steps
    for i in range(6,50, 3):
        print(i)


def double_loop():
    for i in range(0,10):
        for j in range(0,10):
            print(" ({:^2}, {:^2})".format(j,i), end="")
        print()


def menu():
    my_menu = '''
                1: while loop with counter
                2: while loop with quit
                3: for in range
                4: double loop
                0: quit
                '''
    print(my_menu)
    choice = int(input("choose your option: -> "))
    if choice is 1:
        amount = int(input("How many would you like: -> "))
        while_loop_counter(amount)
    elif choice is 2:
        while_loop_indefinite()
    elif choice is 3:
        for_in_range_loop()
    elif choice is 4:
        double_loop()
    elif choice is 5:
        print("Thank you")


menu()
```

Listing 2: Test program - functions and menu

5 Lists

Lists are a collection of data items




```
import random


def get_string(m):
    my_string = input(m)
    return my_string


def get_integer(m):
    my_integer = int(input(m))
    return my_integer


def print_list(L):
    my_string = ""
    for x in L:
        my_string =  my_string + x + " , "
    print(my_string)


def print_with_indexes(L):
    for i in range(0,len(L)):
        output = "{} : {}".format(i, L[i])
        print(output)
```

```
25
26
27 def change_value(L):
28     print_with_indexes(L)
29     index_num = get_integer("Please choose the index value you want to
    change: -> ")
30     new_value = get_string("Please enter the new name: -> ")
31     old_value = L[index_num]
32     L[index_num] = new_value
33     output = "The old value of {} has now been changed to {}".format(
    old_value, L[index_num])
34     print(output)
35
36
37 def remove_item(L):
38     print_list(L)
39     choice = input("Who would you like to remove?  ")
40     if choice in L:
41         L.remove(choice)
42         print("{} has been removed".format(choice))
43     else:
44         print("You choice is not in the list")
45
46
47 def add_to_list(L):
48     print_list(L)
49     choice = input("Who would you like to add?  ")
50     L.append(choice)
51     print("{} has been added".format(choice))
52
53
54 def remove_at_index(L):
55     print_with_indexes(L)
56     choice = int(input("Enter index number to remove:   "))
57     if 0 <= choice < len(L):
58         del L[choice]
59     else:
60         print("This is not a valid index number")
61
62 def insert_at_index(L):
63     choice = int(input("Enter index number to insert at:  ->  "))
64     value = input("Enter value to insert: ->  ")
65     L.insert(choice , value)
66     print_with_indexes(L)
67     return L
68
69
```

```
70 def shuffle_list(L):
71     random.shuffle(L)
72
73
74 def sort_list(L):
75     L.sort()
76
77
78 def find_in_list(L):
79     value = input("Who would you like find? ->  ")
80     if value in L:
81         my_string = "{} is in the list and it is at index #{}".format(value,
       L.index(value))
82     else:
83         my_string = "{} is not in the list".format(value)
84     print(my_string)
85
86
87 def main():
88     names = ["Alice", "Belinda", "Chansing", "Debbie", "Eloise", "Floss"]
89     run = True
90     while run:
91         print("-"*100)
92         menu = {
93             "A" : "Print a list horizontally",
94             "B" : "Print a list vertically with indexes",
95             "C" : "Remove an item from the list",
96             "D" : "Add an item",
97             "E" : "Remove an item at a specified index",
98             "F" : "Insert an item at a given index",
99             "G" : "Sort the list",
100            "H" : "Find an item in the list",
101            "I" : "Shuffle List",
102            "J" : "Change a value",
103            "Q" : "Quit"
104        }
105        i = 0
106        menu_string = ""
107        for x, y in menu.items():
108            menu_string += "| {:2} - {:50} ".format(x,y)
109            i += 1
110            if i != 0 and i%2 == 0:
111                menu_string += "\n"
112        print(menu_string)
113
114        choice = input("Please enter your choice: -> ")
115        print("-" * 100)
```

```
        if choice == "A":
            print_list(names)
        elif choice == "B":
            print_with_indexes(names)
        elif choice ==  "C":
            remove_item(names)
        elif choice == "D":
            add_to_list(names)
        elif choice == "E":
            remove_at_index(names)
        elif choice == "F":
            insert_at_index(names)
        elif choice == "G":
            sort_list(names)
        elif choice == "H":
            find_in_list(names)
        elif choice == "I":
            shuffle_list(names)
        elif choice == "J":
            change_value(names)
        elif choice =="Q":
            print("Thanks , you have quit")
            run = False
        else:
            print("Your entry has not been recognised")


main()
```

Listing 3: Lists

5.1 Issues with lists

- The standard error with lists is: **IndexError: list index out of range**

5.2 What we should be able to do with lists:

- Create one and access items in the list using index notation.
- Find how many items are in the list
- Loop through the list and print out items (have more than one way of doing)
- Understand that there are basic **methods** associated with a list. These mean that we can add, remove, update elements in a list.

5.3 Activities

- Create a function that takes a list as an argument, prints out the list (horizontally), and returns the length of the list
- Create a program that asks for the user to enter 5 numbers, it then prints out the sum, average and product of the numbers (each part is a function).
- Create a program that asks for the user to enter 5 numbers, it then prints a menu asking giving users and option to get the sum, product, average of the numbers

6 Validating User Input

- Create a function to get input of a single letter from a user.
- Create a function to get an integer from a user
- Create a function to get a string between 2 and 5 letters
- Create a function to get an integer from a user with boundaries that might change

7 Looping through lists

8 Lists with more than one dimension

Sometimes we wish to store a "package" of information.
For example , instead of Alice, Bob, Cathy, we might want Alice , Brown, 31 ; Bob Blond 19; Cathy, Black, 21
So we have person, hair colour and age.

```
1  def get_integer(message):
2      my_integer = int(input(message))
3      return my_integer
4
5
6  def get_string(m):
7      my_string = input(m)
8      return my_string
9
10
11 def loop_on_lists(L):
12     print("-" * 30)
13     c = 0
14     for x in L:
```

```
        my_string = "{:2}: {:5} , {:5} , {:^5}".format(c, x[0], x[1], x[2])
        print(my_string)
        c += 1
    print("-" * 30)
    return None


def update_name(L):
    loop_on_lists(L)
    my_index = get_integer("Choose option number for name to update: -> ")
    new_name = get_string("Enter new name: -> ")
    old_name = L[my_index][0]
    L[my_index][0] = new_name
    my_string = "The name for {} has been updated to {}".format(old_name,
   new_name)
    print(my_string)
    return None


def update_hair(L):
    loop_on_lists(L)
    my_index = get_integer("Choose option number for hair to update: -> ")
    new_colour = get_string("Enter new hair colour: -> ")
    L[my_index][1] = new_colour
    my_string = "The hair colour for {} has been updated to {}".format(L[
   my_index][0], L[my_index][1])
    print(my_string)
    return None


def update_age(L):
    loop_on_lists(L)
    my_index = get_integer("Choose option number for age to update: -> ")
    new_age = get_integer("Enter new age: -> ")
    L[my_index][2] = new_age
    my_string = "The age for {} has been updated to {}".format(L[my_index
   ][0], L[my_index][2])
    print(my_string)
    return None


def update(main_list):
    update_menu = [
        ("N", "Name"),
        ("H", "Hair"),
        ("A", "Age"),
        ("B", "Back")
```

```
    ]
    for x in update_menu:
        print("{} : {}".format(x[0], x[1]))
    choice = get_string("Please enter your option letter: -> ")
    if choice is "N":
        update_name(main_list)
    elif choice is "H":
        update_hair(main_list)
    elif choice is "A":
        update_age(main_list)
    elif choice is "B":
        print("Returning to main menu")
        return None
    else:
        print("Unrecognised Entry")
    return None


def search_lists(L):
    search_item = get_string("Please enter your search item: -> ")
    try:
        s = int(search_item)
    except ValueError:
        s = search_item
    row = 0
    for y in L:
        if s in y:
            item_index = y.index(s)
            print("{} has been found at Row {} index {}".format( y[
   item_index],row,  item_index ) )
        row += 1
    return None


def create_new_entry(L):
    name = get_string("Please enter name: -> ")
    hair_colour = get_string("Please enter hair colour: -> ")
    age = get_integer("Please enter age: -> ")
    L.append([name, hair_colour, age])
    return None


def sort_on_index(L, H):
    for i in range(0, len(H)):
        print("{} : {}".format(i, H[i]))
    sort_index = get_integer(" Please choose index to sort on: -> ")
    L = sorted(L, key=lambda x: x[sort_index])
```

```
105     return L
106
107
108 def menu():
109     group_of_lists_header = ["Name", "Hair", "Age"]
110     group_of_lists = [
111         ["Alice", "Blond", 12],
112         ["Cathy", "Blond", 15],
113         ["Dan", "Black", 10],
114         ["Bob", "Brown", 13],
115         ["Joe", "Green", 13]
116     ]
117
118     my_menu = [
119         ("R", "Review"),
120         ("U", "Update"),
121         ("S", "Search"),
122         ("A", "Add new name"),
123         ("O", "Order"),
124         ("Q", "Quit")
125     ]
126
127     run = True
128     while run is True:
129         for x in my_menu:
130             print("{} : {}".format(x[0], x[1]))
131         print("-"*30)
132         choice = get_string("Please enter your option letter: -> ")
133         if choice is "R":
134             loop_on_lists(group_of_lists)
135         elif choice is "U":
136             update(group_of_lists)
137         elif choice is "S":
138             search_lists(group_of_lists)
139         elif choice is "A":
140             create_new_entry(group_of_lists)
141         elif choice is "O":
142             group_of_lists = sort_on_index(group_of_lists,
    group_of_lists_header)
143         elif choice is "Q":
144             print("Thank you")
145             run = False
146         else:
147             print("Unrecognised Entry")
148     return None
149
150
```

```
151 if __name__ == "__main__":
152     menu()
```

Listing 4: Lists

This links to a number of code examples

```
names = ["Alice","Belinda", "Chansing", "Debbie", "Eloise", "Floss"]
# view the list
# view the list with a counter
# find how many elements are in the list
# remove an elements from a list
# access an element in a list


# loop through list
for x in names:
    print(x, end="")
print()
# loop using a counter
for i in range(0, len(names)):
    print(i)
names.remove("Alice")
print(names)
names.append("Alice")
print(names)
names.sort()
print(names)
del names[3]
print(names)
print(names.pop(0))
print(names)
```

Listing 5: Lists

```
/usr/local/bin/python3.7 /Users/Paul/Documents/Python_projects/
   FruitBowlGitHub/lists.py
AliceBelindaChansingDebbieEloiseFloss
0
1
2
3
4
5
['Belinda', 'Chansing', 'Debbie', 'Eloise', 'Floss']
['Belinda', 'Chansing', 'Debbie', 'Eloise', 'Floss', 'Alice']
['Alice', 'Belinda', 'Chansing', 'Debbie', 'Eloise', 'Floss']
['Alice', 'Belinda', 'Chansing', 'Eloise', 'Floss']
Alice
['Belinda', 'Chansing', 'Eloise', 'Floss']
```

Listing 6: Lists

```
def get_numbers():
    num_list = []
```

```
    counter = 0
    while counter < 3:
        num = int(input("Please enter a number "))
        num_list.append(num)
        counter += 1
    num_list.sort()
    print_list(num_list)
    return num_list


def print_list(l):
    my_string = ""
    counter = 0
    for x in l:
        if counter == len(l) - 1:
            my_string = my_string + str(x)
        else:
            my_string = my_string + str(x) + " , "

        counter += 1
    print(my_string)


def add_nums(l):
    _sum = 0
    for x in l:
        _sum += x
    return _sum


def product_nums(l):
    product = 1
    for x in l:
        product = product * x
    return product


def average_nums(l):
    _sum = add_nums(l)
    average = _sum / len(l)
    return average


def print_output(m, v):
    print(30 * "-")
    print("Your {} is :  {}".format(m, v))
    print(30 * "-")
```

```
50
51
52  def run_menu():
53      menu = True
54      menu_list = ["Enter set", "Get average", "Get sum", "Get product"]
55      set = []
56      while menu is True:
57          for i in range(0, len(menu_list)):
58              print("{:<5}{:^10} {:20}".format(i, "--", menu_list[i]))
59          user_choice = input("Please choose your option:  ")
60          if user_choice == "q":
61              return
62          else:
63              user_choice = int(user_choice)
64          if user_choice != 0 and len(set) == 0:
65              print("You have an empty set")
66              continue
67          if user_choice == 0:
68              set = get_numbers()
69          elif user_choice == 1:
70              average = average_nums(set)
71              print_output("average", average)
72          elif user_choice == 2:
73              sum = add_nums(set)
74              print_output("sum", sum)
75          elif user_choice == 3:
76              product = product_nums(set)
77              print_output("product", product)
78          else:
79              print("Unrecognised entry")
80
81
82  if __name__ == "__main__":
83      run_menu()
```

Listing 7: Menu and Functions