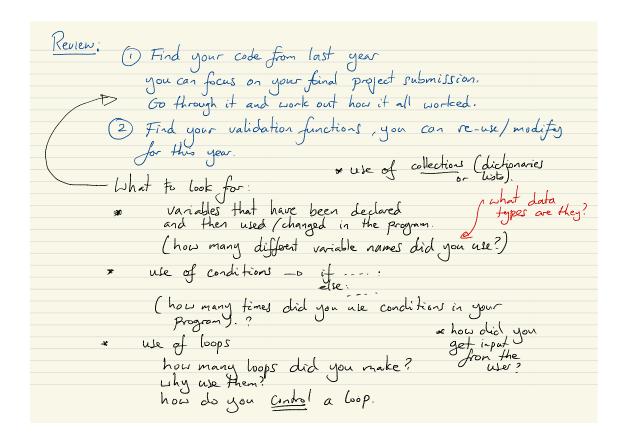
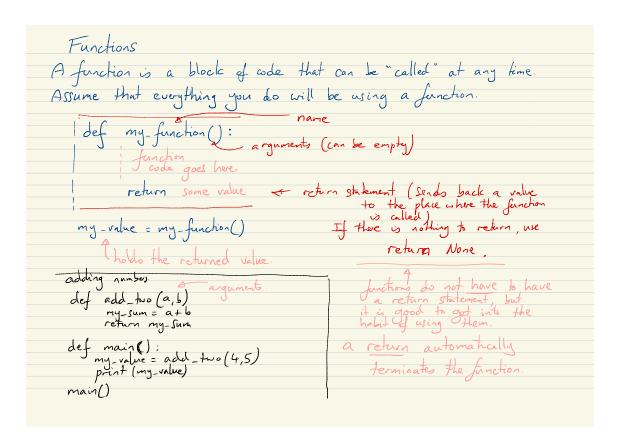
1 Review



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.

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 Lists

Lists are a collection of data items

This links to a number of code examples

4.1 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.

4.2 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

5 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
- \bullet 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

```
names = ["Alice", "Belinda", "Chansing", "Debbie", "Eloise", "Floss"]
2 # view the list
3 # view the list with a counter
4 # find how many elements are in the list
5 # remove an elements from a list
6 # access an element in a list
9 # loop through list
10 for x in names:
print(x, end="")
12 print()
13 # loop using a counter
14 for i in range(0, len(names)):
print(i)
names.remove("Alice")
print(names)
18 names.append("Alice")
19 print(names)
20 names.sort()
21 print(names)
22 del names [3]
23 print(names)
print(names.pop(0))
print(names)
```

Listing 1: Lists

```
1 /usr/local/bin/python3.7 /Users/Paul/Documents/Python_projects/
    FruitBowlGitHub/lists.py
2 AliceBelindaChansingDebbieEloiseFloss
3 0
4 1
5 2
6 3
7 4
8 5
9 ['Belinda', 'Chansing', 'Debbie', 'Eloise', 'Floss']
10 ['Belinda', 'Chansing', 'Debbie', 'Eloise', 'Floss', 'Alice']
11 ['Alice', 'Belinda', 'Chansing', 'Debbie', 'Eloise', 'Floss']
12 ['Alice', 'Belinda', 'Chansing', 'Debbie', 'Floss']
13 Alice
14 ['Belinda', 'Chansing', 'Eloise', 'Floss']
```

Listing 2: Lists

```
names = ["Alice", "Belinda", "Chansing", "Debbie", "Eloise", "Floss"]
```

```
3 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)
12
15 def remove_item(L):
      print_list(L)
16
      choice = input("Who would you like to remove? ")
17
      if choice in L:
          L.remove(choice)
19
          print("{} has been removed".format(choice))
      else:
21
          print("You choice is not in the list")
      print_list(L)
23
25
  def add_to_list(L):
      print_list(L)
      choice = input("Who would you like to add? ")
      L.append(choice)
      print_list(L)
30
  def remove_at_index(L):
32
      print_with_indexes(L)
      choice = int(input("Enter index number to remove: "))
34
      if 0 <= choice < len(L):</pre>
          del L[choice]
          print_with_indexes(L)
      else:
38
          print("This is not a valid index number")
  def sort_list(L):
      L.sort()
42
      print_with_indexes(L)
43
47 #print_list(names)
48 remove_item(names)
49 #print_with_indexes(names)
```

```
# remove_at_index(names)
# add_to_list(names)
# sort_list(names)
```

Listing 3: Lists

```
def get_numbers():
      num_list = []
      counter = 0
      while counter < 3:</pre>
           num = int(input("Please enter a number "))
           num_list.append(num)
           counter += 1
      num_list.sort()
      print_list(num_list)
9
      return num_list
10
11
12
  def print_list(1):
13
      my_string = ""
      counter = 0
      for x in 1:
16
           if counter == len(1) - 1:
17
               my_string = my_string + str(x)
18
           else:
19
               my_string = my_string + str(x) + ", "
20
21
           counter += 1
22
      print(my_string)
23
24
  def add_nums(1):
26
      _sum = 0
      for x in 1:
28
           _sum += x
      return _sum
30
31
32
  def product_nums(1):
33
      product = 1
34
      for x in 1:
35
           product = product * x
      return product
37
  def average_nums(1):
      _sum = add_nums(1)
41
      average = _sum / len(1)
```

```
return average
44
  def print_output(m, v):
46
      print(30 * "-")
      print("Your {} is : {}".format(m, v))
48
      print(30 * "-")
50
  def run_menu():
52
      menu = True
      menu_list = ["Enter set", "Get average", "Get sum", "Get product"]
54
      set = []
      while menu is True:
          for i in range(0, len(menu_list)):
              print("{:<5}{:^10} {:20}".format(i, "--", menu_list[i]))</pre>
          user_choice = input("Please choose your option: ")
59
          if user_choice == "q":
              return
61
          else:
              user_choice = int(user_choice)
63
          if user_choice != 0 and len(set) == 0:
              print("You have an empty set")
65
              continue
          if user_choice == 0:
               set = get_numbers()
          elif user_choice == 1:
               average = average_nums(set)
70
              print_output("average", average)
          elif user_choice == 2:
              sum = add_nums(set)
              print_output("sum", sum)
          elif user_choice == 3:
              product = product_nums(set)
              print_output("product", product)
          else:
              print("Unrecognised entry")
80
82 if __name__ == "__main__":
   run_menu()
```

Listing 4: Menu and Functions