**W3school and real python**

Day 1

* Challenge 1
  + Find the factorial of a number. Take the number from the keyboard as input from the user.
  + Hint : Factorial of 3 = 3 \* 2 \* 1= 6
  + Hint: Try to first find the function from math module and then use it.
* Challenge 2
  + Take the radius of the circle from the keyboard as input from the user and calculate the area and perimeter of it.
  + Hint : Pi \* radius \* radius is the area of circle
  + Hint : 2 \* Pi \* radius is the perimeter of circle
  + Use math module to get the value of Pi ( 3.14 )
* Challenge 3
  + Take first and last name in single command from the user and print  them by replacing the spaces in the string with \* character
* Challenge 4
  + Convert uppercase characters to lowercase and lowercase characters to uppercase ( swap case ) for the name. Take this name as input from the keyboard. Also convert the inputed string in CamelCase or TitleCase.
  + Hint : Try to find some function in the str and see its help on how to use it.
* Challenge 5
  + In a hardcoded string RESTART, replace all the R with $ except the first occurrence and print it.
* Challenge 6
  + Take first and last name in single command from the user and print  them in reverse order with a space between them, find the index using find/index function and then print using slicing concept of the index
  + Hint : Sylvester Fernandes and print Fernandes Sylvester
* Challenge 7
  + Convert the BMI program to use hindi titles while taking input and print weight, height and BMI in Hindi script using formatted strings concept
* **CODE CHALLENGES**
  + Challenge 1
    - Print your First Name and Last Name in Quotation. Both the names should be on different lines
    - Hint : Use the Escape Code for quotation and new line
    - “Sylvester”
    - “Fernandes”
  + Challenge 2
    - Print the temperature of your city in Degree Celsius for the day
    - 26° C
    - Hint : \u00b0 is the unicode for Degree
  + Challenge 2
    - Print the below string
    - Hint :  \u00AE for Registered **®**, \u00A9 for **©** Copyright and \u2122 for TradeMark **™**

**UNIX® and Sun Microsystem™ are  ©, 2018 Oracle**

* + Challenge 3
    - Print your First Name and Last Name in Devanagari Script ( Hindi ) with a comma in between
    - सिल्वेस्टर , फर्नांडीस
    - for help http://jrgraphics.net/r/Unicode/0900-097F
  + Challenge 4
    - Print the checkerboard pattern using escape Codes and Unicode with multiple print statements and the multiplication operator
    - Hint : Eight characters sequence in a line and the next line should start with a space
  + Challenge 5
    - Print the BMI Value Help Screen using the triple quotes syntax
    - World Health Organisation's ( WHO ) BMI VALUES ( 8 Levels )
      * Severe Thinness: less than 16
      * Moderate Thinness: between 16 and 16.9
      * Mild Thinness: between 17 and 18.4
      * Normal: between 18.5 and 24.9
      * Overweight: between 25 and 29.9
      * Obese Class I: between 30 and 34.9
      * Obese Class II: between 35 and 39.9
      * Obese Class III: 40 or greater
* **CODE CHALLENGES**
  + Challenge 1 **( Age Calculator )**
    - Lets assume your age is 21 today. What would be your age after 5 years from today and How old were you 10 years back ?
    - Hint : You need to add to calculate future age and subtract to calculate your past age
  + Challenge 2 **( Height Calculator )**
    - Lets assume your height is 5 Foot and 11 inches, convert 5 Foot into meters and Convert 11 inch into meters and print your total height in meters and centimetres both ?
    - Hint : 1 Foot = 0.3048 meters and 1 inch = 0.0254 meters and 1 m = 100 cm
  + Challenge 3 **( Adult Body Mass Index Calculator )**
    - Assuming your weight ( kilogram ) and height ( meters ), calculate your BMI value and print it ?
    - Hint : Divide your weight in kilograms (kg) by your height in metres (m), then divide the answer by your height again to get your BMI
  + Challenge 4 **( Ponderal Index Calculator  )**
    - Calculate the Ponderal Index of a Person and print it
    - Hint : Divide the BMI by your Height ( meters ) to get your Ponderal Index
  + Challenge 5 **( Target Heart Rate Calculator )**
    - Calculate the Maximum Heart Rate and Target Heart Rate Range ( Lower and Higher value ) of a person of a specific age.
    - Hint : Subtract your age from 220 to get your Maximum Heart Rate
    - Hint : Lower end of your Target Heart Rate is 70% of Maximum Heart Rate
    - Hint : Higher end of your Target Heart Rate is 85% of Maximum Heart Rate
    - Heart Rate = Beats per minute
  + Challenge 6 **( Temperature Calculator )**
    - Assume today's temperature in Jaipur is 29 degree Centigrade. Calculate the temperate in Degree Fahrenheit and print it.
    - Hint : Multiply by 9/5 and add 32
    - Now Calculate the temperature in Degree Kelvin and print it.
    - Hint : Add 273 approximately
  + Challenge 7**( Gas Mileage Calculator )**
    - Assume my car travels 100 Kilometres after putting 5 litres of petrol / Diesel. Calculate the average of my car.
  + Challenge 8 **( Ride Cost Calculator )**
    - Assume you travel 80 km to and fro in a day. Cost of Diesel per litre is 80 INR and your vehicle Fuel Average is 18 km/litre. Now calculate the cost of driving per day to office.
  + Challenge 9 **( Gravity Calculator )**
    - Compute the position of the object after falling for 10 seconds. Output the value meters and assume that the acceleration is -9.81
    - Hint :  Distance = (Acceleration\*Time\*Time ) / 2
  + Challenge 10 **( Weighted Score Calculator )**
    - Lets assume there are 3 assignments and 2 exams, each with max score of 100. Respective weights are 10%, 10%, 10%, 35%, 35% . Compute the weighted score based on individual assignment scores.
    - Hint : weighted score = ( A1 + A2 + A3 ) \*0.1 + (E1 + E2 ) \* 0.35

Day 2

* Code Challenge
  + Challenge 1
    - Remove all the vowels from the list of states

state\_name = [ ‘Alabama’, ‘California’, ‘Oklahoma’, ‘Florida’]

* **Shopping List App**
  + Challenge 1
    - We are going to make a “Shopping List” app.
    - Run the script to start using it.
    - Put new things into the list one at a time
    - Enter the word DONE - in all CAPS - to QUIT the program
    - And once i quit, I want the app to show me everything thats on my list.
* **Shopping List App**
  + Hint 1
    - Step 1: Make a list to hold onto our items.
    - Step 2: Print out instructions on how to use the app
    - Step 3: Ask for new items
    - Step 4: Add new items to our list
    - Step 5: Be able to quit the app
    - Step 6: print out the list
* **Shopping List App** 2
  + Challenge 2
    - If I type SHOW, I should be able to see what is currently in the list
    - If I type HELP, I should be able to see all the help for these special commands
* **Shopping List App** 2
  + Hint 2
    - Step 1: Have a HELP command
    - Step 2: Have a SHOW command
* **Shopping List App** 3
  + Challenge 3
    - User can enter SHOW or Show or show, similar for DONE and HELP, but the program should do the required job
    - Show the item in the list serially starting from 1
    - Let us accept items using a comma separated string
    - Also there should be a functionality to add an item at a specific index
    - There should be a functionality to remove items from the list at a specific index using REMOVE
    - Do all the exception handling for all the extreme use cases

Day 3

* **Hangman Letter Game App**
  + Challenge 1
    - We are going to make a “Hangman Letter” game
    - The computer will pick a word
    - The player can guess it letter by letter or run out of chances.
    - But if they make too many wrong guesses, they loose.
    - If the player makes the right guesses he wins
    - Cleaner interface and option to quit the game
* **Hangman Letter Game App**
  + Hint 1
    - Step 1: Make a list of words, may be Fruits or vegetables
    - Step 2: Pick a random word from the list
    - Step 3: Get a guess from the player
    - Step 4: Compare the guess to the secret number
    - Step 5: If the player guesses the right number print player wins and computer lose.
    - Step 6: If the player guesses the wrong number print player lose and computer wins.
* **Hangman Letter Game App**
  + Challenge 2
    - Screen is messy and rolls ups
    - Convert the code into function

Day 4

* + - Code Challenge to read the zoo.csv file using readlines and print integrating the list
    - Code Challenge to print only the elephant / tiger / lion / zebra / kangaroo
    - Code Challenge to print the total number of water need by elephant / tiger / lion / zebra / kangaroo
    - Code Challenge to print the total number of water needed by all the animals

Day 5

Code Challenges 1

Let’s start with a very simple file of words taken from the text of Romeo and Juliet. (romeo.txt)

We will write a Python program to read through the lines of the file, break each line into a list of words, and then loop through each of the words in the line, and count each word using a dictionary.

Code Challenge 2

Find hash of a file using hashlib library and using SHA-1 algorithm

Code Challenge 3

* + - Find the resolution of any jpeg Image file ( width x height )

Code Challenges 4

With two given lists [1,3,6,78,35,55] and [12,24,35,24,88,120,155], write a program to make a list whose elements are intersection of the above given lists.

Code Challenge 5

With a given list [12,24,35,24,88,120,155,88,120,155], write a program to print this list after removing all duplicate values with original order reserved.

Day 6

Code Challenge 1

Write a program that, given an image file will perform image processing operations.

Your image processing code should make use of PIL library for image processing in Python ([Click Here for Image](http://openedx.forsk.in/c4x/IIIT_Kota/CSP110/asset/sample.jpg)).

Keep only one output image i.e perform all tasks on the same image (override) and print only the name of your output image with extension name in the end of your program.

Take the Image name from User (Handle the extension for image file name in your code)

The image processing features to be provided by your code are:

a.     Greyscale

b.     Rotate\_90 (Rotate the given image file by 90 clockwise)

c.      Crop (Center) (size = 160(W), 204(H))

d.     Thumbnail – Generate the thumbnail of the given image (size = 75, 75)

++++++++++++++++++

Code Challenge 2

Convert all files PNG in a directory into different sizes

Hint: os.listdir(‘.') function will list all the files in the current directory

Code Challenge  
  Name:   
    Regular Expression 1  
  Filename:   
    regex1.py  
  Problem Statement:  
    You are given a string N.   
    Your task is to verify that N is a floating point number.  
    Take Input From User  
  
    In this task, a valid float number must satisfy all of the following   
    requirements:    
     
    Number can start with +, - or . symbol.  
  Hint:   
    Using Regular Expression   
  Input:  
    4    
    4.000  
    -1.00  
    +4.54  
  Output:  
    False  
    True  
    True  
    True  
"""  
  
"""

Code Challenge  
  Name:   
    Regular Expression 2  
  Filename:   
    regex2.py  
  Problem Statement:  
    You are given N email addresses.   
    Your task is to print a list containing only valid email addresses in alphabetical order.  
    Valid email addresses must follow these rules:  
  
    It must have the username@websitename.extension format type.  
    The username can only contain letters, digits, dashes and underscores.  
    The website name can only have letters and digits.  
    The minimum length is 2 and maximum length of the extension is 4.   
  Hint:   
    Using Regular Expression   
  Input:  
    [lara@hackerrank.com](mailto:lara@hackerrank.com)  
    [brian-23@hackerrank.com](mailto:brian-23@hackerrank.com)  
    [britts\_54@hackerrank.com](mailto:britts_54@hackerrank.com)  
  Output:  
    ['[brian-23@hackerrank.com](mailto:brian-23@hackerrank.com)', '[britts\_54@hackerrank.com](mailto:britts_54@hackerrank.com)', '[lara@hackerrank.com](mailto:lara@hackerrank.com)']  
  
"""  
  
"""  
  
Code Challenge  
  Name:   
    Regular Expression 3  
  Filename:   
    regex3.py  
  Problem Statement:  
    You and Virat are good friends.   
    Yesterday, Virat received credit cards from ABCD Bank.   
    He wants to verify whether his credit card numbers are valid or not.   
    You happen to be great at regex so he is asking for your help!  
  
    A valid credit card from ABCD Bank has the following characteristics:  
  
    It must start with a '4', '5' or ' 6'.  
    It must contain exactly 16 digits  
    It must only consist of digits (0-9)  
    It may have digits in groups of 4, separated by one hyphen "-"  
    It must NOT use any other separator like ', ' , '\_', etc  
    It must NOT have 4 or more consecutive repeated digits   
   
  Hint:   
    Using Regular Expression   
  Input:  
    4123456789123456  
    5123-4567-8912-3456  
    61234-567-8912-3456  
    4123356789123456  
    5133-3367 -8912-3456  
    5123 - 3567 - 8912 - 3456  
  Output:  
    Valid  
    Valid  
    Invalid  
    Valid  
    Invalid  
    Invalid  
"""  
  
"""  
  
Code Challenge  
  Name:   
    Regular Expression 4  
  Filename:   
    regex4.py  
  Problem Statement:  
    You are given email addresses.   
    Your task is to print a list containing only valid email addresses in lexicographical order .  
    (Take input from User)  
  
    Valid email addresses must follow these rules:  
  
    It must have the username@websitename.extension format type.  
    The username can only contain letters, digits, dashes and underscores.   
    The website name can only have letters and digits.   
    The maximum length of the extension is    
   
  Hint:   
    Using Regular Expression   
  Input:  
    [ajeet@forsk.com](mailto:ajeet@forsk.com)  
    [kunal-23@forsk.com](mailto:kunal-23@forsk.com)  
    [yogendra\_54@forsk.com](mailto:yogendra_54@forsk.com)  
  Output:  
    ['[ajeet@forsk.com](mailto:ajeet@forsk.com)', '[kunal-23@forsk.com](mailto:kunal-23@forsk.com)', '[yogendra\_54@forsk.com](mailto:yogendra_54@forsk.com)’]  
  
"""  
  
"""  
  
Code Challenge  
  Name:   
    Post Codes UK  
  Filename:   
    postcodes\_uk.py  
  Problem Statement:  
    We write an expression, which is capable of recognizing the postal codes or postcodes of the UK.  
  
    Postcode units consist of between five and seven characters,   
    which are separated into two parts by a space.   
      
    The two to four characters before the space represent the so-called outward   
    code or out code intended to direct mail from the sorting office to the delivery office.   
      
    The part following the space, which consists of a digit followed by two uppercase characters,   
    comprises the so-called inward code, which is needed to sort mail at the final delivery office.   
      
    The last two uppercase characters do not use the letters CIKMOV,   
    so as not to resemble digits or each other when hand-written.  
  
    The outward code can have the form: One or two uppercase characters,   
    followed by either a digit or the letter R,   
    optionally followed by an uppercase character or a digit.   
    (We do not consider all the detailed rules for postcodes,   
    i.e only certain character sets are valid depending on the position   
    and the context.)      
  Hint:   
    Using Regular Expression   
  Input:  
      example\_codes = ["SW1A 0AA", # House of Commons  
                 "SW1A 1AA", # Buckingham Palace  
                 "SW1A 2AA", # Downing Street  
                 "BX3 2BB", # Barclays Bank  
                 "DH98 1BT", # British Telecom  
                 "N1 9GU", # Guardian Newspaper  
                 "E98 1TT", # The Times  
                 "TIM E22", # a fake postcode  
                 "A B1 A22", # not a valid postcode  
                 "EC2N 2DB", # Deutsche Bank  
                 "SE9 2UG", # University of Greenwhich  
                 "N1 0UY", # Islington, London  
                 "EC1V 8DS", # Clerkenwell, London  
                 "WC1X 9DT", # WC1X 9DT  
                 "B42 1LG", # Birmingham  
                 "B28 9AD", # Birmingham  
                 "W12 7RJ", # London, BBC News Centre  
                 "BBC 007" # a fake postcode  
                ]  
  
"""  
  
"""  
  
Code Challenge  
  Name:   
    Post Codes Germany  
  Filename:   
    postcodes\_germany.py  
  Problem Statement:  
    We have to bring together the information of two files.   
    In the first file, we have a list of nearly 15000 lines of post codes with the   
    corresponding city names plus additional information.   
  
    The other file contains a list of the 19 largest German cities.   
    Each line consists of the rank, the name of the city, the population,   
    and the state (Bundesland):   
      
    Our task is to create a list with the top 19 cities,  
    with the city names accompanied by the postal code.   
    If you want to test the following program,   
    you have to save the list above in a file called largest\_cities\_germany.txt   
    and you have to download and save the list of German post codes  
  
  Output:  
    The output of this file looks like this,   
    but we have left out all but the first three postal codes for every city:   
      
        Berlin {'10715', '13158', '13187', ...}  
        Hamburg {'22143', '22119', '22523', ...}  
        München {'80802', '80331', '80807', ...}  
        Köln {'51065', '50997', '51067', ...}  
        Frankfurt am Main {'65934', '60529', '60308', ...}  
        Essen {'45144', '45134', '45309', ... }  
        Dortmund {'44328', '44263', '44369',...}  
        Stuttgart {'70174', '70565', '70173', ...}  
        Düsseldorf {'40217', '40589', '40472', ...}  
        Bremen {'28207', '28717', '28777', ...}  
        Hannover {'30169', '30419', '30451', ...}  
        Duisburg {'47137', '47059', '47228', ...}  
        Leipzig {'4158', '4329', '4349', ...'}  
        Nürnberg {'90419', '90451', '90482', ...}  
        Dresden {'1217', '1169', '1324', ...}  
        Bochum {'44801', '44892', '44805', ...}  
        Wuppertal {'42109', '42119', '42287', ...}  
        Bielefeld {'33613', '33607', '33699', ...}  
        Mannheim {'68161', '68169', '68167', ...}  
        
  Hint:   
    Using Regular Expression   
    zuordnung\_plz\_ort.csv  
    list of nearly 15000 lines of post codes with the corresponding city names   
    plus additional information.  
  
"""  
  
# Add a challenge for german postal code  
# file already present in data folder   
# post\_codes\_germany.txt