Data Science Fundamentals: Python | Table of Contents (../index.ipynb)

Module 3. Control Flow (./01_mod_control_flow.ipynb) | Statements (./02_python_statements.ipynb) | Functions (./03_python_functions.ipynb) | **Exercises (./04_mod_exercises.ipynb)**

Module 3: Practice Exercises

1. Guess a number between 1 to 9 given by the user. Hint. Random (https://docs.python.org/3/library/random.html)

In [1]:
```
import random

target_num, guess_num = random.randint(1, 10), 0
while target_num != guess_num:
    guess_num = int(input('Guess a number between 1 and 10 until you get it ri
ght : '))
print('You guessed correct!')
```

```
Guess a number between 1 and 10 until you get it right : 5
Guess a number between 1 and 10 until you get it right : 3
Guess a number between 1 and 10 until you get it right : 2
Guess a number between 1 and 10 until you get it right : 8
You guessed correct!
```

1. Check the validity of password input by the user. Hint. Re (https://docs.python.org/3/library/re.html) which does matching operations much like Perl.

Validation:

- At least 1 letter between [a-z] and 1 letter between [A-Z].
- At least 1 number between [0-9].
- At least 1 character from [$#@].
- Minimum length 6 characters.
- Maximum length 16 characters.

In [4]:
```
import re
p = input("Input your password : ")
x = True
while x:
    if (len(p)<6 or len(p)>12):
        break
    elif not re.search("[a-z]",p):
        break
    elif not re.search("[0-9]",p):
        break
    elif not re.search("[A-Z]",p):
        break
    elif not re.search("[$#@]",p):
        break
    elif re.search("\s",p):
        break
    else:
        print("Valid Password")
        x = False
        break

if x:
    print("Not a Valid Password")
```

```
Input your password : Be1B@now
Valid Password
```

1. Get input of the age of 3 people by user and determine oldest and youngest among them

In [5]:
```
number1 = int(input("Enter First Person's Age : "))
number2 = int(input("Enter Second Person's Age : "))
number3 = int(input("Enter Third Person's Age : "))

def largest(num1, num2, num3):
    if (num1 > num2) and (num1 > num3):
        largest_num = num1
    elif (num2 > num1) and (num2 > num3):
        largest_num = num2
    else:
        largest_num = num3
    print("The Oldest of All Three People is : ", largest_num)

def smallest(num1, num2, num3):
    if (num1 < num2) and (num1 < num3):
        smallest_num = num1
    elif (num2 < num1) and (num2 < num3):
        smallest_num = num2
    else:
        smallest_num = num3
    print("The Youngest Of All Three People is : ", smallest_num)

largest(number1, number2, number3)
smallest(number1, number2, number3)
```

```
Enter First Person's Age : 34
Enter Second Person's Age : 37
Enter Third Person's Age : 46
The Oldest of All Three People is :  46
The Youngest Of All Three People is :  34
```

1. A student will not be allowed to sit in exam if his/her attendance is less than 75%.

Take following input from user

- Number of classes held
- Number of classes attended.
- And print percentage of class attended
- Is student is allowed to sit in exam or not?

In [6]:
```
#number of classes held:
total = int(input("How many clases are held : "))

#number of classes attended:
attended = int(input("How many classes have you attended : "));

#percentage of classes
percentage = ((attended/total)*100);

if percentage < 75:
    print("You can not take part in the exam");
else:
    print("You are free to take part in the exam");
```

```
How many clases are held : 20
How many classes have you attended : 17
You are free to take part in the exam
```

1. Get an integer N from the user and perform the following actions:

Find out -

- if N is odd, print "weird"
- if N is even and in the inclusive range of 2 to 5, print "Not Weird"
- if N is even and in the inclusive range of 6 to 20, print "Weird"
- if N is even and greater than 20, print "Not Weird"

In [7]:
```
# Given an integer, n, perform the following conditional actions:
# If n is odd, print Weird
# If n is even and in the inclusive range of 2 to 5, print Not Weird
# If n is even and in the inclusive range of 6 to 20, print Weird
# If n is even and greater than 20, print Not Weird

N = int(input("Input a Number to see if You Are Weird or Not Weird : "))

if N % 2 != 0:
    print("Weird")
else:
    if N >= 2 and N <= 5:
        print("Not Weird")
    elif N >= 6 and N <= 20:
        print("Weird")
    elif N > 20:
        print("Not Weird")
```

```
Input a Number to see if You Are Weird or Not Weird : 23
Weird
```

Extra Points

1. ***Optional Exercise (Extra Points)***. Write a Python program to reverse a string.

Sample String : "1234abcd" Expected Output : "dcba4321"

In [ ]:
```
x =
```

In [8]:
```
def string_reverse(str1):

    rstr1 = ''
    index = len(str1)
    while index > 0:
        rstr1 += str1[ index - 1 ]
        index = index - 1
    return rstr1
#print(string_reverse('1234abcd'))

r = str(input('Input a string to reverse: '))
string_reverse(r)
```

```
Input a string to reverse: 2468good
```

Out[8]:
```
'doog8642'
```

1. ***Optional Exercise (Extra Points)***. Write a Python function to multiply all the numbers in a list.

Sample List : (8, 2, 3, -1, 7) Expected Output : -336

In [14]:
```
def multiply(numbers):
    total = 1
    for x in numbers:
        total *= x
    return total
print(multiply((8, 2, 3, -1, 7)))
```

```
-336
```

1. ***Optional Exercise (Extra Points)***. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.

Sample String : 'The quick Brow Fox' Expected Output : No. of Upper case characters : 3 No. of Lower case Characters : 12

In [9]:
```
def string_test(s):
    d={"UPPER_CASE":0, "LOWER_CASE":0}
    for c in s:
        if c.isupper():
           d["UPPER_CASE"]+=1
        elif c.islower():
           d["LOWER_CASE"]+=1
        else:
           pass
    print ("Original String : ", s)
    print ("No. of Upper case characters : ", d["UPPER_CASE"])
    print ("No. of Lower case Characters : ", d["LOWER_CASE"])

# string_test('The quick Brown Fox')
x = str(input('Input a sentence to calculate the number of upper and lower cas
e letters: '))
string_test(x)
```

```
Input a sentence to calculate the number of upper and lower case letters: Dat
a Science is intriguing
Original String :  Data Science is intriguing
No. of Upper case characters :  2
No. of Lower case Characters :  21
```