

Module 01

Algorithms

Data Science Developer

FizzBuzz

- Fizz Buzz is an algorithm function that will log out to the console every number from 1 to “num”.
- For each number, if the number is divisible by 3, it'll log out the word “*Fizz*” instead of that number.
- Next, if the number is divisible by 5, it'll log out the word “*Buzz*” instead of that number.
- And finally, if a number is divisible by both 3 and 5, we want to logout the word “*FizzBuzz*” instead of that number.
- Beside is the result of *fizzBuzz(20)*

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
```

FizzBuzz

```
def fizzBuzz(num) :  
    for i in range(1,num+1) :  
        if (i % 15 == 0) :  
            print('FizzBuzz');  
        elif (i % 3 == 0) :  
            print('Fizz');  
        elif (i % 5 == 0) :  
            print('Buzz');  
        else :  
            print(i);  
  
fizzBuzz(20);
```

Fibonacci

Fibonacci sequence characterized by the fact that every number after the first two is the sum of the two preceding ones:

1, 1, 2, 3, 5, 8, 13, 21,
34, 55, 89, 144...

input= **fibonacci(6)** then output= 8

Fibonacci

```
def fibo(urut) :  
    listData = [1,1];  
    for i in range(2,urut):  
        listData.append(listData[i-2] + listData[i-1]);  
    return listData[urut-1];  
  
print(fibo(8));
```

Reverse List In Place

This algorithm function will take in a list as a parameter, then it'll reverse that list and return us the reversed list.

`reverseList([1,2,3,4,5,6,7,8])`

1,2,3,4,5,6,7,8



8,7,6,5,4,3,2,1

Reverse List In Place

```
import math;

def reverseList(theList) :
    for i in range(0, math.floor(len(theList)/2)) :
        tempList = theList[i];
        theList[i] = theList[len(theList) - 1 - i];
        theList[len(theList) - 1 - i] = tempList;

    return theList;

print(reverseList([1,2,3,4,5,6,7,8]));
```

Bubble Sort

```
x = [6000, 34, 203, 3, 746, 200, 984, 198, 764, 9, 1]
```

```
def bubbleSort(list) :  
    for i in range(len(list), 0, -1) :  
        for j in range(0,i-1) :  
            if (list[j] > list[j + 1]) :  
                temp = list[j];  
                list[j] = list[j + 1];  
                list[j + 1] = temp;  
    return list;  
  
print(bubbleSort(x));
```


Mean, Median & Mode

- **Mean** is the average value of a dataset.
- **Median** is the middle number of a dataset.
- **Mode** is the most frequent number of a dataset.

[1,2,3,2,5,2,7,2]

Mean = 3 | Median = 2 | Mode = 2

Mean

```
x = [ 1,2,3,2,5,2,7,2 ]
```

```
def getMean(list) :  
    sum = 0;  
    for item in list :  
        sum += item;  
  
    mean = sum / len(list);  
    return mean;  
  
print(getMean(x));
```

Median

```
x = [ 1,2,3,2,5,2,7,2 ]
```

```
def getMedian(list) :  
    list.sort();  
    median = 0;  
    if (len(list) % 2 != 0) :  
        median = list[floor(len(list) / 2)];  
    else :  
        mid1 = list[(int(len(list) / 2)) - 1];  
        mid2 = list[int(len(list) / 2)];  
        median = (mid1 + mid2) / 2;  
    return median;  
  
print(getMedian(x));
```

```
x = [ 1,2,3,2,5,2,7,2 ]
```

```
def getMode(list) :  
    countList = [];  
    # create countList  
    for num in list :  
        check = False;  
        for list1 in countList :  
            if(list1[0] == num) :  
                list1[1] += 1;  
                check = True;  
        if(check == False) :  
            countList.append([num, 0]);  
  
    # create list of mode/s  
    maxFrequency = 0;  
    modes = [];  
    for list1 in countList :  
        if (list1[1] > maxFrequency) :  
            modes = [list1[0]];  
            maxFrequency = list1[1];  
        elif (list1[1] == maxFrequency) :  
            modes.append(list1[0]);  
  
    # if every value appears same amount of times  
    if (len(modes) == len(countList)) :  
        modes = [];  
    return modes;  
  
print(getMode(x));
```

Mode