

3rd Feb Assignment

February 7, 2023

1 Assignment 4

Q1. Which keyword is used to create a function? Create a function to return a list of odd numbers in the range of 1 to 25.

Ans. def keyword is used to create a function

```
[2]: ## function to return a list of odd numbers in the range of 1 to 25.  
def func1():  
    n=[]  
    for i in range(1,26):  
        n.append(i)  
    return n
```

```
[3]: func1()
```

```
[3]: [1,  
      2,  
      3,  
      4,  
      5,  
      6,  
      7,  
      8,  
      9,  
      10,  
      11,  
      12,  
      13,  
      14,  
      15,  
      16,  
      17,  
      18,  
      19,  
      20,  
      21,  
      22,
```

```
23,  
24,  
25]
```

Q2. Why `*args` and `**kwargs` is used in some functions? Create a function each for `*args` and `**kwargs` to demonstrate their use.

Ans. We use `*args` and `**kwargs` when we are unsure about number of arguments

```
[4]: #function for *args  
def func2(*args):  
    return args
```

```
[5]: func2(1,2,3,4, 'Jp', True, 3.6)
```

```
[5]: (1, 2, 3, 4, 'Jp', True, 3.6)
```

```
[6]: #function for *kwargs  
def func3(**kwargs):  
    return kwargs
```

```
[7]: func3(a=1,b=2,c=3)
```

```
[7]: {'a': 1, 'b': 2, 'c': 3}
```

Q3. What is an iterator in python? Name the method used to initialise the iterator object and the method used for iteration. Use these methods to print the first five elements of the given list `[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]`.

Ans. An iterator in Python is an object that implements the iterator protocol, which consists of two methods: `iter()` and `next()`. The `iter()` method is used to initialize the iterator object, and the `next()` method is used for iteration.

```
[8]: class IterateList:  
    def __init__(self, lst):  
        self.lst = lst  
        self.index = 0  
  
    def __iter__(self):  
        return self  
  
    def __next__(self):  
        if self.index >= len(self.lst):  
            raise StopIteration  
        else:  
            result = self.lst[self.index]  
            self.index += 1
```

```

        return result

lst = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
iter_lst = IterateList(lst)

for i in range(5):
    print(next(iter_lst))

```

2
4
6
8
10

Q4.What is generator function in python? Why yield keyword is used? Give an example of generator function?

Ans.A generator function in Python is a special type of function that returns a generator iterator, which is an object that can be iterated over to produce a sequence of values. Unlike a normal function, a generator function does not return a value when it is called. Instead, it returns a generator object that can be used to produce a sequence of values.

The yield keyword is used in a generator function to produce a value and temporarily suspend the function's execution, so that it can be resumed later from where it left off.

```

[9]: #generator function to return fibonaaci series
def test_fib(n):
    a,b=0,1
    for i in range(n):
        yield a #it gives a record one by one
        a,b=b,a+b

```

```

[11]: for i in test_fib(10):
        print(i)

```

0
1
1
2
3
5
8
13
21
34

Q5. Create a generator function for prime numbers less than 1000. Use the next() method to print the first 20 prime numbers.

```
[28]: def prime_number():  
    primes=[] #storing the prime numbers  
    for num in range(2,1000):  
        for prime in primes:  
            if num%prime==0:  
                break  
        else:  
            primes.append(num)  
            yield num
```

```
[29]: prime=prime_number()
```

```
[30]: for i in range(20):  
    print(next(prime))
```

```
2  
3  
5  
7  
11  
13  
17  
19  
23  
29  
31  
37  
41  
43  
47  
53  
59  
61  
67  
71
```