Excercise_2_solutions

May 30, 2019

1 LECTURE 2

Construct a generator which returns a sequence of prime numbers as a function, class and expression ().

1st as a function

In [1]:
```
# def myprimef():
#   n = 2
#   while True:
#     isprime = True

#     for i in range(2, n):
#       if n == 2:
#         yield n

#       if n % i == 0:
#         isprime = False
#         break

#     if isprime:
#       yield n

#     n += 1

# prime = myprimef()
# print(next(prime))
# print(next(prime))
# print(next(prime))
# print(next(prime))
# print(next(prime))
# print(next(prime))
# print(next(prime))

# another topic
def myprimef(n):
  for i in range(2, n + 1):
    if all(i % j != 0 for j in range(2, i)):
```

```
            yield i

        print(list(myprimef(22)))
```

```
[2, 3, 5, 7, 11, 13, 17, 19]
```

1st as a class

```
In [2]: class MyPrimeC(object):
          def __init__(self, n):
            self.n = n
            self.i = 2
          def __next__(self):
            while True:
              i = self.i
              self.i += 1

              if i > self.n:
                return

              if all(i % j != 0 for j in range(2,i)):
                return i

        prime = MyPrimeC(33)
        print(next(prime))
        print(next(prime))
        print(next(prime))
        print(next(prime))
        print(next(prime))
        print(next(prime))
```

```
2
3
5
7
11
13
```

1st as a expression

```
In [3]: primeg = lambda n: (i for i in range(2, n + 1) if all(i % j != 0 for j in range(2, i)))
        print(list(primeg(44)))
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43]
```

Fill in the Line class methods to accept coordinate as a pair of tuples and return the slope and distance of the line.

2nd problem

```
In [4]: import math
        class Line:
            def __init__(self,coor1,coor2):
                """Initialize instance attributes with tuples (x1,y1)  and (x2,y2)
                """
                self.coor1 = coor1
                self.coor2 = coor2
                self.slp = round((coor1[1]-coor2[1])/(coor1[0]-coor2[0]),2)
                self.dist = math.sqrt((coor1[0]-coor2[0])**2 + (coor1[1]-coor2[1])**2)


            def distance(self):
                """Calculate the length of the segment (line)
                """
                return self.dist

            def slope(self):
                """ Return the slope of a line going through the ends ( the 'a' in y=ax+b)
                """
                return self.slp

        cr1 = (1,1)
        cr2 = (4,5)
        line1 = Line(cr1,cr2)
        line1.distance()
        line1.slope()
```

```
Out[4]: 1.33
```

3rd problem
Fill in the class

```
In [5]: class Cylinder(object):

            def __init__(self,height=1,radius=1):
                self.height = height
                self.radius = radius
                self.vol = round(math.pi*self.radius**2*self.height,2)
                self.surfarea = round(2*math.pi*self.radius**2 + 2*math.pi*self.radius*self.hei

            def volume(self):
                return self.vol

            def surface_area(self):
                return self.surfarea


        c = Cylinder(2,3)
```

```
        print(c.volume())
        print(c.surface_area())
```

```
56.55
94.2
```

```
In [6]:     def volume(self):
                print(f"{self.vol:.2f}")

            def surface_area(self):
                print(f"{self.surfarea:.1f}")
```

4th problem

```
In [7]: class DataFile(object):

          def __init__(self, filename='undef'):
            (...)

          def info(self):
            (...)

          def avg(self, colnum=0, colname=''):
            """ The column name or colnum can be provided alternatively
            """
            (...)

          def min(self, colnum=0, colname=''):
              (...)

          def max(self, colnum=0, colname=''):
              (...)

          def distinc(self, colnum=0, colname=''):
             "Counts distinct number of values in a given column."
              (...)


          File "<ipython-input-7-c594afb2aad0>", line 22
        (...)
        ^
    IndentationError: unexpected indent
```

2 New Section