Shantanu Joshi #073

Day 0

```
log = input('Line')
print('Hello World Shantanu')
print('Hello World')
```

Day 1

```
for i in [1, 2, 3, "abc"]:
    log_call = input(f"Enter the Number {i}")
    log = int(log_call)
    log_total = log + int("Enter Number")
print(log_total)
```

Day 2

```
import numpy as np

x = 5
y = 5
log = x + y
print(log)
```

Day 3

```
a = int(input("Enter a"))
if a > 0 and a % 2 == 0:
    print("Positive Even")
else:
    print("Other")
```

Day 4

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def greet(self):
        print(f"Hello {self.name}")

    def walk(self):
        print(f"{self.name} is walking")

p1 = Person("Shantanu")
p1.greet()
p1.walk()
```

Day 5

```
for i in range(1, 6):
    print(i)
```

Day 6

```
nums = [1, 2, 3, 4, 5]
square = [n*n for n in nums]
print(square)
```

Day 7

```
a = [1, 2, 3, 4, 5, 6, 7, 8, 9]
for i in a:
    print(i)
```

Day 8

```
def check(num):
    if num > 0:
        return "positive"
    elif num == 0:
        return "zero"
    else:
        return "negative"
print(check(5))
```

Day 9

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
print(factorial(5))
```

Day 10

```
x = 10
y = 3
print(x + y)
print(x - y)
print(x * y)
```

Day 11

```
data = [10, 20, 30, 40]
total = sum(data)
avg = total / len(data)
print(total)
print(avg)
```

Day 12

```
class Animal:
    def __init__(self, name):
        self.name = name

    def sound(self):
        pass

class Dog(Animal):
    def sound(self):
        return "Bark"

class Cat(Animal):
    def sound(self):
        return "Meow"
```

Day 13

```
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author

    def info(self):
        return f"{self.title} by {self.author}"
```

Day 14

```
class FileManager:
    def __init__(self, filename):
        self.filename = filename

    def read(self):
        with open(self.filename) as f:
            return f.read()
```

Day 15

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def insert(self, data):
        new_node = Node(data)
        new_node.next = self.head
        self.head = new_node
```

Day 16

```
try:
    x = int(input())
except:
    print("Error")
```

Day 17

```
def add(a, b):
    return a + b
print(add(2, 3))
```

Day 18

```
class Calculator:
    def __init__(self):
        self.result = 0

    def add(self, x):
        self.result += x

    def subtract(self, x):
        self.result -= x
```

Day 19

```
import random
print(random.randint(1, 10))
```

Day 20

```
numbers = [1, 2, 3, 4, 5]
even = [n for n in numbers if n % 2 == 0]
print(even)
```

Day 21

```
class Stack:
    def __init__(self):
        self.items = []

    def push(self, item):
        self.items.append(item)

    def pop(self):
        return self.items.pop()
```

Day 23

```
def fibonacci(n):
    a, b = 0, 1
    for i in range(n):
        print(a)
        a, b = b, a + b
```

Day 24

```
class Queue:
    def __init__(self):
        self.items = []

    def enqueue(self, item):
        self.items.append(item)

    def dequeue(self):
        return self.items.pop(0)
```

Day 25

```
import math
print(math.sqrt(16))
print(math.pi)
```

Day 26

```
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, n):
        if n % i == 0:
            return False
    return True
```

Day 27

```
class BankAccount:
    def __init__(self, balance=0):
        self.balance = balance

    def deposit(self, amount):
        self.balance += amount

    def withdraw(self, amount):
        self.balance -= amount
```

Day 28

```
def reverse_string(s):
    return s[::-1]
print(reverse_string("hello"))
```

Day 29

```
import sys
for arg in sys.argv:
    print(arg)
```