

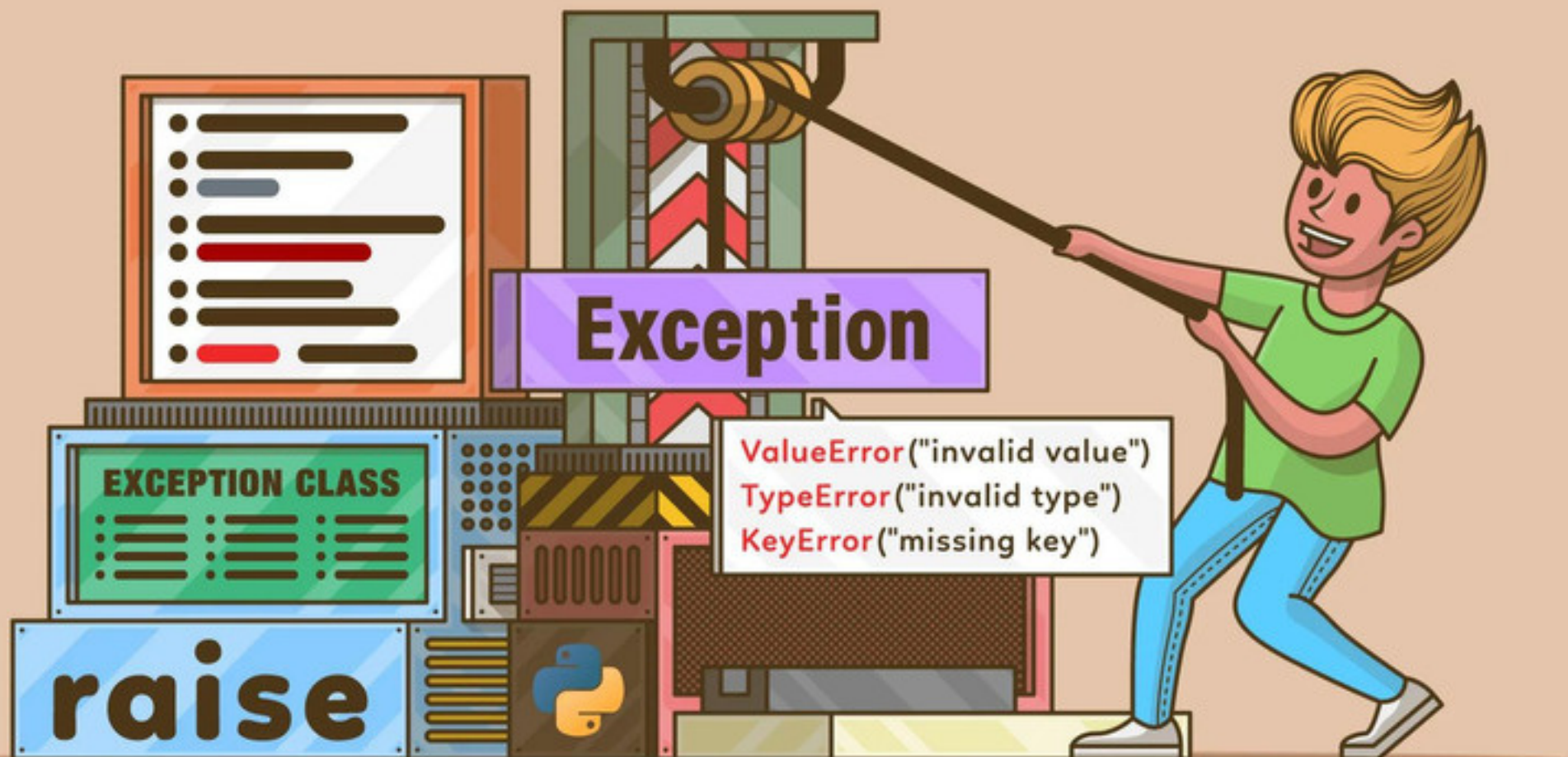
#DAY

23

# DAY 23 OF 200 DAY'S PYTHON CHALLENGE



## EXCEPTION HANDLING



# EXCEPTION HANDLING

Exception handling in Python is a mechanism for handling errors that occur during the execution of a program. It allows the programmer to specify what should happen when an error occurs, rather than the program simply crashing.

There are two main ways to handle exceptions in Python: using the “Try” and “Except” statements

We are learn two more statements in Exception Handling :  
One is Else or another one is Finally

- **Try:**

We call a try statement the block of code and lines in which the error occurs is expected.

- **Except:**

Perform except statement if any type of error occurs in try block

- **Else:**

This block execute only when try block execute successfully

- **Finally:**

The finally block in Python is always executed, regardless of whether an exception is raised in the try block or not.

# TRY-EXCEPT-ELSE



python.py

```
try :  
    f= open("test.txt" , 'r')  
except Exception as e :  
    print("this is my except block " , e)
```



python.py

```
try :  
    f= open("test.txt" , 'w')  
    f.write(" write into my file " )  
except Exception as e :  
    print("this is my except block " , e)  
else :  
    f.close()  
    print(" this will be executed once your try will  
execute without error" )
```

# TRY-EXCEPT-FINALLY



python.py

```
try :  
    f = open("test3.txt" , 'w')  
    f.write("write something" )  
finally :  
    print("finally will execute itself in any situation")
```



python.py

```
try :  
    f = open("test3.txt" , 'r')  
    f.write("write something" )  
finally :  
    print("finally will execute itself in any situation")
```

# CUSTOM EXCEPTION HANDLING



python.py

```
age = int(input("enter your age" ))
class validateage(Exception):
    def __init__(self , msg) :
        self.msg = msg

def validaetage(age) :
    if age < 0 :
        raise validateage("entered age is negative " )
    elif age > 200 :
        raise validateage("entered age is very very high " )
    else :
        print("age is valid" )

try :
    age = int(input("enter your age" ))
    validaetage(age)
except validateage as e :
    print(e)
```

# ASSIGNMENT QUESTIONS

**Q1. What is an Exception in python? Write the difference between Exceptions and syntax errors.**

**Q2. What happens when an exception is not handled? Explain with an example.**

**Q3. Which python statement are used to catch and handle exceptions? Explain with an example.**

**Q4. Explain with an example:**

- try and else
- finally
- raise

**Q5. What are Custom Exceptions in python? Why do we need custom Exceptions? Explain with an example.**

**Q6. Create a custom exception class. Use this class to handle an exception.**

# APRIORI ANALYSIS

Apriori Analysis  $\rightarrow$  Approximate Results

Time complexity  $\rightarrow$  order of magnitude of a statement.

$\rightarrow$  no. of times any statement is executing

Example 1:

main()  
 $x = y + 2 \rightarrow \textcircled{1} \Rightarrow O(1)$  Asymptotic Notations  
 $\hookrightarrow$  Big O (Worst case time complexity)

Example 2:

main()  
 $x = y + 2 \xrightarrow{\textcircled{1}} i < n$   
for  $i$  in range(0, n):  
 $\downarrow$   
Loop  $x = y + 2$   
 $i = 0 \quad i = 1 \quad \dots \quad i = n-1$   
 $x = y + 2 \quad x = y + 2 \quad x = y + 2$   
 $\textcircled{n}$   
 $n+1$  times  $\hookrightarrow O(n)$   
 $n = 1000000$

Example 3:

main()  
 $x = y + 2 \xrightarrow{\textcircled{1}}$   
for ( $i = 0; i < n; i++$ )  
 $\{$   
 $\quad x = y + 2 \xrightarrow{\textcircled{n}}$   
 $\}$   
for ( $i = 0; i < n; i++$ )  
 $\{$   
 $\quad \text{for } (j = 0; j < n; j++)$   
 $\quad \{$   
 $\quad \quad x = y + 2 \xrightarrow{\textcircled{n^2}}$   
 $\quad \}$   
 $\}$   
 $\}$

$n = 1000000$   
 $1 + n + n^2 \hookrightarrow O(n^2)$   
 $O(n^2) > O(n) > O(1)$   
 (constant)

$i = 0$	$j = 0 \quad j = 1 \quad j = 2 \quad \dots \quad j = n-1$	$i = 1$	$j = 0 \quad j = 1 \quad j = 2 \quad \dots \quad j = n-1$
$x = y + 2$		$x = y + 2$	

$i < n \Rightarrow i = n-1$   
 $j = 0 \quad j = 1 \quad \dots \quad j = n-1$   
 $x = y + 2$

$n = 6$

# APRIORI ANALYSIS

Apriori Analysis

Example 4

```

main()
{
    i = n;
    while (i > 1)
    {
        i = i - 1;
    }
}
    
```

$n = 5$   
 $i = 5$   $5 > 1$   $i = 4$   
 $i = 4$   $4 > 1$   $i = 3$   
 $i = 3$   $3 > 1$   $i = 2$   
 $i = 2$   $2 > 1$   $i = 1$   
 $i = 1$   $1 > 1$  — **False** (Not Execute)

$n = 5$  — 4 times  
 $n = 10$  — 9 times  
 $n$  —  $(n-1)$  times

$O(n-1) \approx O(n)$

Example 5

```

main()
{
    i = n;
    while (i > 1)
    {
        i = i - 5;
    }
}
    
```

$n = 10$   $i = 10$   
 $10 > 1$   $i = 5$   $5 > 1$   $i = 0$   
 $5 > 1$   $i = 0$   
 $0 > 1$  (False)

$n = 10$  — 5 times  
 $n = 100$  — 50 times  
 $n$  —  $n/5$  times

$O(n/5) \approx O(n)$

Note

- 1) Time complexity is loop only
- 2) Higher loop  $\rightarrow n^2 + n + 1 \rightarrow O(n^2)$
- 3) No loop at all  $\rightarrow$  constant time complexity  $\rightarrow O(1)$



# APRIORI ANALYSIS

Time complexity (Apriori Analysis)

Example 6

```

main()
{
  i = n
  while (i > 2)
    i = i1/2
}
    
```

$n = 256$  — 3 times

$n = 256$  — True  
 $256 > 2$  — True  
 $i = (256)^{1/2} = (2^8)^{1/2} = 2^4 = 16$   
 $16 > 2$  — True  
 $i = (16)^{1/2} = (2^4)^{1/2} = 2^2 = 4$   
 $4 > 2$  — True  
 $i = (4)^{1/2} = 2$   
 $2 > 2$  — False

$\log_a a = 1$   
 $\log_a n^b = b \log_a n$   
 $\log_2 n^{1/2^k} = 1$   
 $\log$  on both sides  
 $\frac{1}{2^k} \log_2 n = \log_2 2 = 1$   
 $\log_2 n = 2^k$   
 $\log_2 (\log_2 n) = k \log_2 2 = k$   
 $\log_2 (\log_2 n) = k$

$i = n$   
 $n^{1/2}$   
 $(n^{1/2})^{1/2} = n^{1/4}$   
 $((n^{1/2})^{1/2})^{1/2} = n^{1/8}$   
 $\downarrow$  k times  
 $n^{1/2^k}$

$n^{1/2^k} = 2$

$\log_2$  on both sides;  
 $\log_2 n^{1/2^k} = \log_2 2 = 1$   
 $\frac{1}{2^k} \log_2 n = 1$   
 $\log_2 n = 2^k$   
 $\log_{25} (\log_2 n) = k \log_{25} 2$   
 $k = \log_{25} (\log_2 n)$

$O(\log_2 (\log_2 n))$

Example 7

```

main()
{
  i = n
  while (i > 2)
    i = i1/25
}
    
```

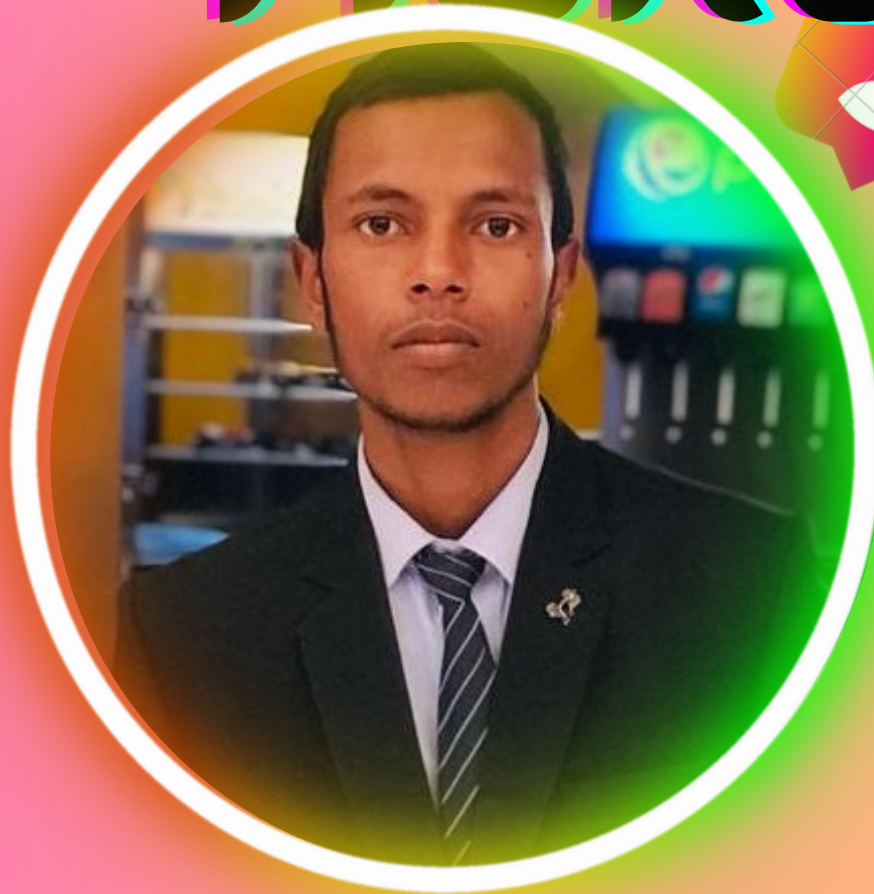
$n = 256$  — True

$n^{1/25^k} = 2$   
 $\log_2$  on both sides;  
 $\log_2 n^{1/25^k} = \log_2 2 = 1$   
 $\frac{1}{25^k} \log_2 n = 1$   
 $\log_2 n = 25^k$   
 $\log_{25} (\log_2 n) = k \log_{25} 2$   
 $k = \log_{25} (\log_2 n)$

$\log_{25} (\log_2 n) = k \log_{25} 2$   
 $k = \log_{25} (\log_2 n)$

$O(\log_{25} (\log_2 n))$

# FOLLOW FOR MORE



**FOLLOW ME**

