```
'''
Starting at 9:05 pm
'''
```

```
# Introduction to Tuples
planets=["mercury","venus","earth","mars","jupiter","saturn","uranus","neptune"]
planets.append("Sun")
print(planets)
```

```
['mercury', 'venus', 'earth', 'mars', 'jupiter', 'saturn', 'uranus', 'neptune
```

```
a=(1,2,3,4,5)
print(type(a))
```

```
<class 'tuple'>
```

```
b=()
print(type(b))
```

```
<class 'tuple'>
```

```
c=(1) # integer 1 in brackets
print(type(c))
d=(1,) #tuple containing single number 1
print(type(d))
```

```
<class 'int'>
<class 'tuple'>
```

```
# Creating Tuples with the range Keyword
a=tuple(range(10))
print(a)
ar=[1,2,3,4,5]
ar[0]=10
print(ar)
a[0]=10 #tuple is immutable
```

```
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
[10, 2, 3, 4, 5]
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-1238815979.py in <cell line: 0>()
      5 ar[0]=10
      6 print(ar)
----> 7 a[0]=10 #tuple is immutable

TypeError: 'tuple' object does not support item assignment
```

Next steps: Explain error

```
# Tuple containing a mutable list
tup=(1,2,3,['a','b']) #4 elements
print(tup)
tup[3].append('c')
print(tup)
```

```
(1, 2, 3, ['a', 'b'])
(1, 2, 3, ['a', 'b', 'c'])
```

```
#Quiz 1
t = (1, 2, 3)
t[0] = 4
print(t)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-2612507583.py in <cell line: 0>()
      1 #Quiz 1
      2 t = (1, 2, 3)
----> 3 t[0] = 4
      4 print(t)

TypeError: 'tuple' object does not support item assignment
```

Next steps: Explain error

```
#Quiz 2
a = 23
t = (a) # t=(23) -> single number is in bracket
print(type(t)) #int
tu=(a,)
print(type(tu))
ti=("hi")
print(type(ti))
```

```
<class 'int'>
<class 'tuple'>
<class 'str'>
```

```
#Strings in Python
# anything -> '' or ""
a="abc"
b='abc'
print(type(a))
print(type(b))

#ASCII
# a - z : 97 to 122
# a - z : 97 to 122
# A - Z : 65 to 90
# '0' - '9' : 48 to 57
```

```
<class 'str'>
<class 'str'>
```

```
# ord
print(ord('B'))
print(ord('b'))
print(ord('1'))
# chr
print(chr(91))
print(chr(89))
```

```
66
98
49
[
Y
```

```
a="Scaler Academy"
print(a[0]) #accessing is allowed
a[0]='s'
#strings are immutable in nature
```

```
S
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-3229008962.py in <cell line: 0>()
      1 a="Scaler Academy"
      2 print(a[0]) #accessing is allowed
----> 3 a[0]='s'

TypeError: 'str' object does not support item assignment
```

Next steps: Explain error

```
#homogenity
result = a+42
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-1578736774.py in <cell line: 0>()
      1 #homogenity
----> 2 result = a+42

TypeError: can only concatenate str (not "int") to str
```

Next steps: Explain error

```
#iterability

for char in a: # a ="Scaler Academy"
  print(char)
```

```
S
c
a
l
e
r

A
c
a
d
e
m
y
```

```
#case sentivity
b="scaler academy"
# a ="Scaler Academy"
print(a==b)

print("ABCD"=="ABCD")
```

```
False
True
```

```
# Quiz 3
# Which one of the following is a valid string?
'''
    •   "ScaLer#' -> invalid " '
    •   %adfa" -> missing "
    •   "^&abc#" -> "" valid
    •   'academy -> ' missing
```

```
#Quiz 4
s = 'Scaler Academy'
# S - 0 c - 1 a - 2 l - 3 e - 4 r - 5
print(s[0:5]) #start=0, end=5-1 = 4
```

```
Scale
```

```
#Quiz 5
a = '1' #str
b = '2' #str
c = a + b #str+str -> concatenation
print(c)
```

```
12
```

```
#Quiz 6
a = 'abcd'
a += 'e' # "abcd" + "e" -> "abcde"
print(len(a)) #len("abcde")
```

```
5
```

```
#Slicing in Strings
text ="ScalerAcademy"

print(text[0:6]) # s=0, e=5 Scaler
print(text[-3:]) # s=-3, end=last character of the string emy
print(text[6:11]) #s=6, e=10, Acade
print(text[::-1]) # ymedacArelacS
print(text[::2]) # SaeAaey
print(text[3:]) #lerAcademy
```

```
Scaler
emy
Acade
ymedacArelacS
SaeAaey
lerAcademy
```

```
# break : 10 pm - 10:10 pm
```

```
#Functions in Strings

#capitalize() -> capitalizes only first letter
print('hi i am akanksha, im a educator'.capitalize())

#title() -> capitalizes first letter of each word
print("hi i am akanksha, im a educator".title())

#upper() -> convert entire string to uppercase
print("hi i am akanksha, im a educator".upper())
```

```
Hi i am akanksha, im a educator
Hi I Am Akanksha, Im A Educator
HI I AM AKANKSHA, IM A EDUCATOR
```

```
#count(substring) -> counts the occurences of the substring in the string
print("pooja makhija".count("ja"))
```

```
 print("AAAaaaAAAaaaAAAaaa".count("AAA"))
```

```
2
3
```

```
#replace(old,new) -> replace old occurence with a new substring
print("Ketan Dongare".replace('e','o'))
```

```
Kotan Dongaro
```

```
#split(separator)
a="Smitha ,Guna ,Raja"
print(a.split(','))
```

```
['Smitha', 'Guna', 'Raja']
```

```
# join() -> concatenate using a specified character
#pass a list containing string
separator='.'
result=separator.join(["hi",'there'])
print(result)

r="@".join(["2","31","56"])
print(r)

re="@".join([2,31,56])
print(re)
```

```
hi.there
2@31@56
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-368206090.py in <cell line: 0>()
      7 print(r)
      8 
----> 9 re="@".join([2,31,56])
     10 print(re)

TypeError: sequence item 0: expected str instance, int found
```

Next steps: Explain error

```
#Formatted Strings
name="Mahesh"
gender="Male"
age=21

print("Name: - ",name, "Gender: - ",gender,"Age: - ", age)
```

```
Name: -  Mahesh Gender: -  Male Age: -  21
```

```
template="Name: - {}, Gender: - {}, Age: -{}"
print(template.format(name,gender,age))
```

```
Name: - Mahesh, Gender: - Male, Age: -21
```

```
#List Comparison
[1,2,3,4,5] < [1,3]
# 1 < 1 -> False
# 2 < 3 -> True
```

```
True
```

```
[1,3,0] > [1,3]
# 1 > 1 -> false
# 3 > 3 -> false
# 0 extra element True
```

```
True
```

```
[1,3] <[1,2,3,4,5]
```

```
False
```

```
[1, 3] == [1, 3]
```

```
True
```

```
#String comparison -> ascii values based

'a'>'A'
```

```
True
```

```
'Raja' < "Hari"
```

```
False
```

```
#Print the longest log entry
'''
Given a list of strings where each string
represents a line of text,
find and print the longest string from the list.
'''
example
05-09-2025 22:44:00 Server started


lines = [
    "DevOps is awesome.",
    "Devops is awesome!",
    "Error: Disk not found.",
    "Server started successfully at 10:05 AM.",
    "OK"
]

longest_line="" #length=18
for line in lines:
  if len(line)>len(longest_line): #22>18
    longest_line=line #Error: Disk not found.

print(longest_line) #Server started successfully at 10:05 AM.
```

```
Server started successfully at 10:05 AM.
```