Class 4 Notes

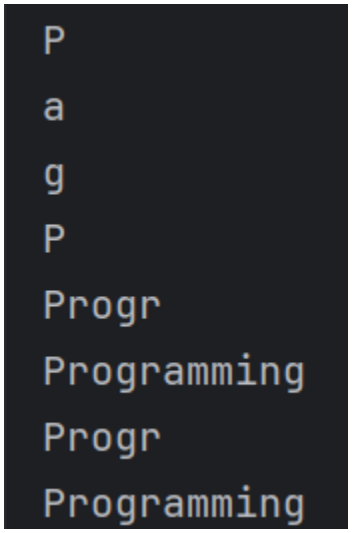
How to find an address of a variable

Program	Output
`name ="Programming"` `print(name)` `print(id(name))` `print(hex(id(name)))`	`Programming` `2274768447536` `0x211a2c3e830`

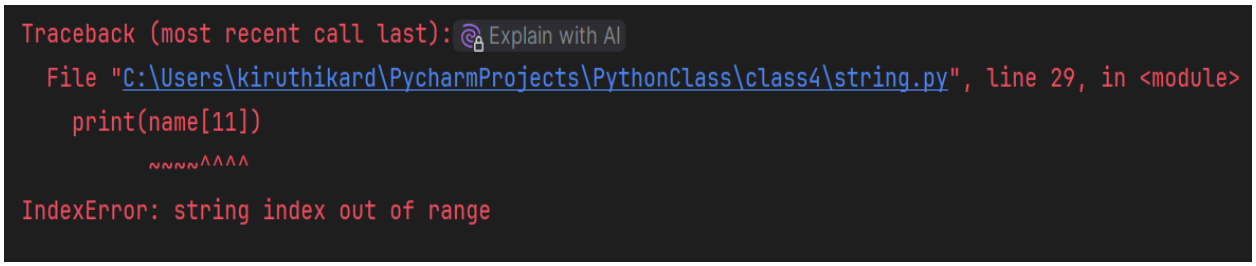
Python is case sensitive . In the below table you may noticed that same address for value ‘hi’ and different address for ‘HI’ and ‘hello’

Program	Output
`` myname = 'hi' print(id(myname)) myName = 'hi' print(id(myName)) ```	``` 140712768081744 140712768081744 ``` For this refer above memory picture
`` myname='HI' print(id(myname)) myName = 'hi' print(id(myName)) ```	``` 2822283574512 140712768081744 ``` Here different address. Case sensitive
`` myname = 'hi' print(id(myname)) myName = 'hello' print(id(myName)) ```	``` 140712768081744 2822283574176 ``` Different values, different address

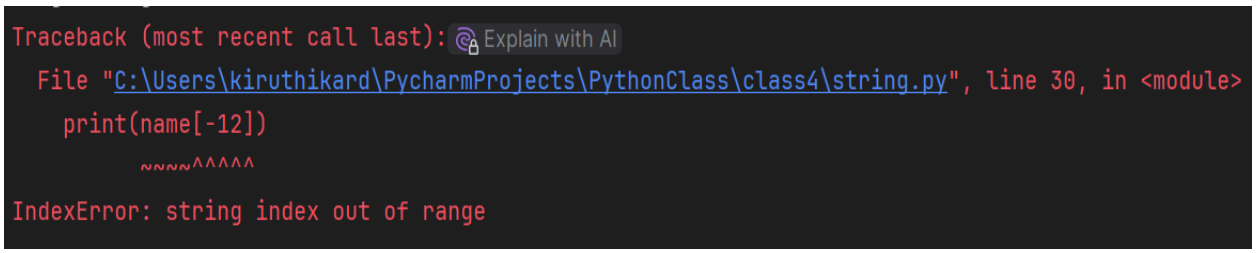
How to Access Elements in a String

Program	Output
	
print(name[11])	Error (picture 1)
print(name[-12])	Error (picture 2)

Picture 1



Picture 2



Escape Sequences

```
#Escape sequences
# \"
#\'
#\\
#\n

statement = 'My dog's name is Daisy'  #error
print(statement)
```

```
C:\Users\kiruthikard\AppData\Local\Programs\Python\Python313\python.exe C:\Users\kiruthikar
  File "C:\Users\kiruthikard\PycharmProjects\PythonClass\class4\string.py", line 38
    statement = 'My dog's name is Daisy'  #error
                                  ^
SyntaxError: unterminated string literal (detected at line 38)

Process finished with exit code 1
```

To solve above error there are two ways as below

```
statement = "My dog's name is Daisy"
print('statement: ' + statement)

statement1 = 'My dog\'s name Daisy'
print('statement1: '+ statement1)
```

```
statement3 = "My dog"s name Daisy"
```

```
C:\Users\kiruthikard\AppData\Local\Programs\Python\Python313\python.exe C:\Users\kiruthikard\
  File "C:\Users\kiruthikard\PycharmProjects\PythonClass\class4\string.py", line 46
    statement3 = "My dog"s name Daisy"
                                ^
SyntaxError: unterminated string literal (detected at line 46)

Process finished with exit code 1
```

```
statement4 = 'My dog"s name Daisy'
print('statement4: ' + statement4)
statement5 = "My dog\"s name Daisy"
print('statement5: ' + statement5)
```

```
statement4: My dog"s name Daisy
statement5: My dog"s name Daisy
```

String Methods

```
14    #len()
15    #To find the length of the string
16    name ="Programming"
17    print(len(name))
18    print('length of the value in name variable is: ' + str(len(name)))
19    print('length of the value in name variable is: ' + (len(name)))
20
```

```
11
length of the value in name variable is: 11
Traceback (most recent call last):  Explain with AI
  File "C:\Users\kiruthikard\PycharmProjects\PythonClass\class4\string.py", line 19, in <module>
    print('length of the value in name variable is: ' + (len(name)))
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~^~~~~~~~~~~~~
TypeError: can only concatenate str (not "int") to str

Process finished with exit code 1
```

```python
#String functions
# python everything is an object. dot notation to access
#the method

name ="programming in python"
print(name.upper())
print(name.lower())
print(name.title())
print(name.capitalize())
name_space ="  Programming in python   "
print(name_space)
print(name_space.strip())#remove white space beginning & end
print(name_space.lstrip())
print(name_space.rstrip())
name_space ="  Programming in python   "
print(name_space.find("gra")) #return index
#python is case sensitive language
print(name_space.find("Gra"))
print(name_space.replace( __old: "m", __new: "p"))
print("in" in name_space) # return boolean
print("hello" not in name_space)
```

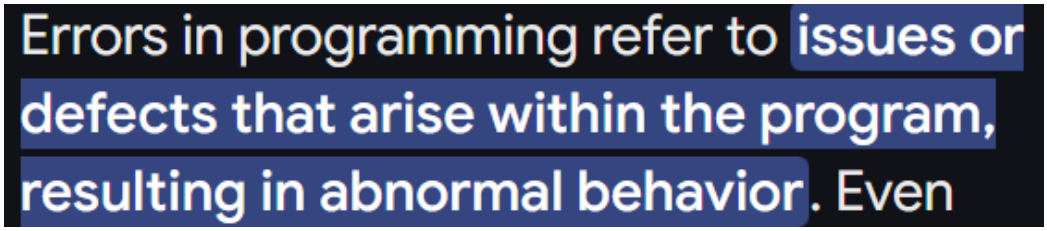
Output

```
PROGRAMMING IN PYTHON
programming in python
Programming In Python
Programming in python
  Programming in python
Programming in python
Programming in python
  Programming in python
5
-1
  Prograpping in python
True
True
False
```

Errors

Errors will give you information about

1. Why the program execution is stooped
2. Why it didn't produce the expected output
3. Why it's behaving in a abnormal way



In Error messages, you can find which line number is causing issue and what kind of error and why the error occurred. By reading the error messages you can fix the issue and execute the program. For example,

```
C:\Users\kiruthikard\AppData\Local\Programs\Python\Python313\python.exe C:\Users\kiruthikard\P
  File "C:\Users\kiruthikard\PycharmProjects\PythonClass\class4\string.py", line 46
    statement3 = "My dog"s name Daisy"
                                    ^
SyntaxError: unterminated string literal (detected at line 46)

Process finished with exit code 1
```

In the above error message, you can find

- the line number causing issue here it is ‘your project folder structure/filename/line 46’.
- Name of the error here it is ‘SyntaxError’
- Detailed Information about the error here it is ‘unterminated string literal(detected at line46)’. It means it’s expecting starting and closing double quotes

Error

Syntax Error

Syntax errors happen when your code does not make any sense to the computer. This can happen because you've misspelt something or there's too many brackets or a missing comma.

```
print(12 + 4))
  File "<stdin>", line 1
    print(12 + 4))
                  ^
SyntaxError: unmatched ')'
```

Name Error

This happens when there is a variable with a name that the computer does not recognise. It's usually because you've misspelt the name of a variable you created earlier.
Note: variable names are case sensitive!

```
my_number = 4
my_Number + 2
Traceback (most recent call
last): File "<stdin>", line 1,
NameError: name 'my_Number'
is not defined
```

Zero Division Error

This happens when you try to divide by zero, This is something that is mathematically impossible so Python will also complain.

```
5 % 0
Traceback (most recent call last): File "<stdin>", line 1, ZeroDivisionError: integer division or modulo by zero
```