In [1]:
```
print(3 + 2) # addition(+)
```
```
5
```

In [2]:
```
print(3 - 2) # subtraction(-)
```
```
1
```

In [3]:
```
print(3 * 2) # multiplication(*)
```
```
6
```

In [4]:
```
print(3 / 2) # division(/)
```
```
1.5
```

In [5]:
```
print(3 ** 2) # exponential(**)
```
```
9
```

In [6]:
```
print(3 % 2) # modulus(%)
```
```
1
```

In [7]:
```
print(3 // 2) # Floor division operator(//)
```
```
1
```

In [ ]:
```
```

checking data types

In [8]:
```
print(type(10)) # int
```
```
<class 'int'>
```

In [9]:
```
print(type(3.14)) # float
```
```
<class 'float'>
```

In [10]:
```
print(type(1 + 3j)) # complex
```
```
<class 'complex'>
```

In [11]:
```
print(type('samsonkadarikota')) # string
```
```
<class 'str'>
```

In [14]:
```
print(type([1, 2, 3])) # List
```
```
<class 'list'>
```

In [15]:
```
print(type({'name':'samson'})) # Dictionary
```
```
<class 'dict'>
```

In [16]:
```
print(type({9.8, 3.14, 2.7})) # Set
```

```
<class 'set'>
```

In [17]:
```
print(type((9.8, 3.14, 2.7))) # Tuple
```

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

In [18]:
```
print(type(3 == 3)) # Bool
```

```
<class 'bool'>
```

In [19]:
```
print(type(3 >= 3)) # Bool
```

```
<class 'bool'>
```

In [ ]:
```

```

Single line comment

In [20]:
```
letter = 'p' # A string could be a single character or a bunch of texts
```

In [21]:
```
print(letter) # P
```

```
p
```

In [22]:
```
print(len(letter)) # l
```

```
1
```

In [23]:
```
greeting = 'Hello, World!' # String could be  a single or double quote,"Hello, Worl
```

In [24]:
```
print(greeting) # Hello, World!
```

```
Hello, World!
```

In [25]:
```
print(len(greeting)) # 13
```

```
13
```

In [26]:
```
sentence = "I hope you are enjoying 30 days of python challenge"
```

In [27]:
```
print(sentence)
```

```
I hope you are enjoying 30 days of python challenge
```

In [ ]:
```

```

Multiline String

In [33]:
```
multiline_string = '''I am a teacher and enjoy teaching.
I didn't find anything as rewarding as empowering people.
That is why I created 30 days of python.'''
```

In [34]:
```
print(multiline_string)
```

```
I am a teacher and enjoy teaching.
I didn't find anything as rewarding as empowering people.
That is why I created 30 days of python.
```

In [ ]:

Another way of doing the same thing

In [35]:
```
multiline_string = """I am a teacher and enjoy teaching.
I didn't find anything as rewarding as empowering people.
That is why I created 30 days of python."""
print(multiline_string)
```

```
I am a teacher and enjoy teaching.
I didn't find anything as rewarding as empowering people.
That is why I created 30 days of python.
```

In [ ]:

String Concatenation

In [36]:
```
first_name = 'Asabeneh'
last_name = 'Yetayeh'
space = ' '
full_name = first_name  +  space + last_name
print(full_name) # Asabeneh Yetayeh
# Checking length of a string using len() builtin function
print(len(first_name))  # 8
print(len(last_name))   # 7
print(len(first_name) > len(last_name)) # True
print(len(full_name)) # 15
```

```
Asabeneh Yetayeh
8
7
True
16
```

In [ ]:

Unpacking characters

In [37]:
```
language = 'Python'
a,b,c,d,e,f = language # unpacking sequence characters into variables
print(a) # P
print(b) # y
print(c) # t
print(d) # h
print(e) # o
print(f) # n
```

```
P
y
t
h
o
n
```

In [ ]:

Accessing characters in strings by index

In [38]:
```
language = 'Python'
first_letter = language[0]
print(first_letter) # P
second_letter = language[1]
print(second_letter) # y
last_index = len(language) - 1
last_letter = language[last_index]
print(last_letter) # n
```

```
P
y
n
```

In [ ]:

If we want to start from right end we can use negative indexing. -1 is the last index

In [39]:
```
language = 'Python'
last_letter = language[-1]
print(last_letter) # n
second_last = language[-2]
print(second_last) # o
```

```
n
o
```

In [ ]:

Slicing

In [40]:
```
language = 'Python'
first_three = language[0:3] # starts at zero index and up to 3 but not include 3
last_three = language[3:6]
print(last_three) # hon
# Another way
last_three = language[-3:]
print(last_three)   # hon
```

```
last_three = language[3:]
print(last_three)   # hon
```

```
hon
hon
hon
```

In [ ]:

Skipping character while splitting Python strings

In [41]:
```
language = 'Python'
pto = language[0:6:2]
print(pto)
```

```
Pto
```

In [ ]: