

Basic caliculations

```
In [2]: print(3 + 2)    # addition(+)
print(3 - 2)    # subtraction(-)
print(3 * 2)    # multiplication(*)
print(3 / 2)    # division(/)
print(3 ** 2)   # exponential(**)
print(3 % 2)    # modulus(%)
print(3 // 2)   # Floor division operator(//)
```

```
5
1
6
1.5
9
1
1
```

```
In [ ]:
```

Checking data types

```
In [1]: print(type(10))
print(type(3.14))
print(type(1 + 3j))
print(type('sunitha'))
print(type([1, 2, 3]))
print(type({'name':'sunitha'}))
print(type({9.8, 3.14, 2.7}))
print(type((9.8, 3.14, 2.7)))
print(type(3 == 3))
print(type(3 >= 3))
```

```
<class 'int'>
<class 'float'>
<class 'complex'>
<class 'str'>
<class 'list'>
<class 'dict'>
<class 'set'>
<class 'tuple'>
<class 'bool'>
<class 'bool'>
```

```
In [ ]:
```

single line comment

```
In [4]: letter = 'P'
print(letter)
```

```
P
```

```
In [5]: print(len(letter))
```

```
1
```

```
In [ ]:
```

```
In [6]: greeting = 'Hello, World!'
print(greeting)
```

```
Hello, World!
```

```
In [7]: print(len(greeting))
```

```
13
```

```
In [ ]:
```

```
In [8]: sentence = "Success usually comes to those who are too busy to be looking for it
print(sentence)
```

```
Success usually comes to those who are too busy to be looking for it.
```

```
In [9]: print(len(sentence))
```

```
69
```

```
In [ ]:
```

Multiline String

```
In [13]: multiline_string='''I joined Naresh IT
for Agentic AI course'''
print(multiline_string)
```

```
I joined Naresh IT
    for Agentic AI course
```

```
In [ ]:
```

```
In [16]: multiline_string="""I joined Naresh IT
for Agentic AI course"""
print(multiline_string)
```

```
I joined Naresh IT
    for Agentic AI course
```

```
In [ ]:
```

String Concatenation

```
In [ ]:
```

```
In [1]: first_name='sunitha'
second_name='venu'
space=' '
```

```
full_name=first_name+space+second_name  
print(full_name)
```

sunitha venu

In []:

```
first_name='sunitha'  
second_name='venu'  
full_name=first_name+" "+second_name  
print(full_name)
```

sunitha venu

In []:

Checking length of a string using len() builtin function

```
In [3]: print(len(first_name))  
print(len(second_name))  
print(len(first_name) > len(second_name))  
print(len(full_name))
```

7
4
True
12

In []:

Unpacking characters

```
In [21]: language = 'Python'  
a,b,c,d,e,f = language  
print(a)  
print(b)  
print(c)  
print(d)  
print(e)  
print(f)
```

P
y
t
h
o
n

In []:

Accessing characters in strings by index

```
In [4]: language='PYTHON'
first_letter=language[0]
print(first_letter)
```

P

```
In [5]: second_letter=language[1]
print(second_letter)
```

Y

```
In [7]: last_index=len(language)-1
last_letter=language[last_index]
print(last_letter)
```

N

```
In [8]: last_letter=language[last_index]
print(last_letter)
```

N

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

```
In [10]: language="SUNITHA"
last_letter=language[-1]
print(last_letter)
```

A

```
In [ ]: 
```

```
In [11]: second_last=language[-2]
print(second_last)
```

H

```
In [ ]: 
```

slicing

```
In [12]: name="sunitha"
first_three=name[0:3]
print(first_three)
```

sun

```
In [ ]: 
```

```
In [13]: last_three=name[3:6]
print(last_three)
```

ith

```
In [16]: last_three=name[-3:]
print(last_three)
```

tha

```
In [18]: last_three=name[3:]
print(last_three)
```

itha

```
In [ ]:
```

```
In [19]: print('I hope every one enjoying the python challenge.\nDo you ?') # Line break
print('Days\tTopics\tExercises')
print('Day 1\t3\t5')
print('Day 2\t3\t5')
print('Day 3\t3\t5')
print('Day 4\t3\t5')
print('This is a back slash symbol (\\"))' # To write a back slash
print('In every programming language it starts with \"Hello, World!\"')
```

I hope every one enjoying the python challenge.

Do you ?

Days	Topics	Exercises
Day 1	3	5
Day 2	3	5
Day 3	3	5
Day 4	3	5

Days	Topics	Exercises
Day 1	3	5
Day 2	3	5
Day 3	3	5
Day 4	3	5

This is a back slash symbol (\")

In every programming language it starts with "Hello, World!"

String Methods

capitalize(): Converts the first character the string to Capital Letter

```
In [21]: challenge="thirty days of python"
print(challenge.capitalize())
```

Thirty days of python

```
In [ ]:
```

count():returns occurrences of substring in string

```
In [2]: challenge='thirty days of python'
print(challenge.count('y'))
print(challenge.count('y',7,14))
print(challenge.count('th'))
```

3

1

2

```
In [ ]:
```

endswith():checks if a string ends with a specified ending

```
In [3]: challenge='thirty days of pyhton'
print(challenge.endswith('on'))
print(challenge.endswith('tion'))
```

True
False

In []:

expandtabs(): Replaces tabs with spaces.

```
In [4]: challenge='thirty\tdays\tot\tpyhton'
print(challenge.expandtabs())
print(challenge.expandtabs(10))
```

thirty days of pyhton
thirty days of pyhton

In []:

find(): returns the index of first occurrence of substring

```
In [7]: challenge='thirty days of pyhton'
print(challenge.find('y'))
print(challenge.find('th'))
```

5
0

In []:

format():format string into nicer output

```
In [12]: first_name='sunita'
last_name='venu'
job='Gen AI developer'
country='India'
sentence='I am {} {}. I am a {}. I live in {}'.format(first_name, last_name, job, country)
print(sentence)
```

I am sunita venu. I am a Gen AI developer. I live in India

In []:

```
In [18]: radius=10  
pi=3.14  
area=pi*radius**2  
result='the area of circle with {} is {}'.format(str(radius),str(area))  
print(result)
```

```
the area of circle with 10 is 314.0
```

```
In [ ]:
```

isalnum(): check alphanumeric character

```
In [21]: challenge='ThirtyDaysPython'  
print(challenge.isalnum())
```

```
True
```

```
In [ ]:
```

```
In [22]: challenge='30DaysPython'  
print(challenge.isalnum())
```

```
True
```

```
In [ ]:
```

```
In [23]: challenge='Thirty Days Python'  
print(challenge.isalnum())
```

```
False
```

```
In [ ]:
```

```
In [24]: challenge='thirty Days Python 2019'  
print(challenge.isalnum())
```

```
False
```

```
In [ ]:
```

isalpha(): checks if all characters are alphabets

```
In [29]: challenge='ThirtyDaysofPython'  
print(challenge.isalpha())
```

```
True
```

```
In [ ]:
```

```
In [30]: num='123'  
print(num.isalpha())
```

```
False
```

In []:

isdigit(): check digit character

```
In [31]: challenge='Thirty'  
print(challenge.isdigit())
```

False

In []:

```
In [32]: challenge='30'  
print(challenge.isdigit())
```

True

In []:

isdecimal() : checks decimal characters

```
In [33]: num='10'  
print(num.isdecimal())
```

True

```
In [35]: num='10.5'  
print(num.isdecimal())
```

False

In []:

isidentifier(): checks for valid identifier

```
In [36]: challenge='30days of python'  
print(challenge.isidentifier())
```

False

```
In [37]: challenge='Thirty_days_of_python'  
print(challenge.isidentifier())
```

True

In []:

islower(): checks if all alphabets in a strings are lowercase

```
In [39]: challenge='ThirtyDaysPython'  
print(challenge.islower())
```

```
False
```

```
In [ ]:
```

```
In [43]: challenge='thirty days of python'  
print(challenge.islower())
```

```
True
```

```
In [ ]:
```

isupper():

```
In [1]: challenge='thirty days of python'  
print(challenge.isupper())
```

```
False
```

```
In [ ]:
```

```
In [2]: challenge='THIRTY DAYS OF PYTHON'  
print(challenge.isupper())
```

```
True
```

```
In [ ]:
```

```
In [3]: challenge='Thirty Days Of Python'  
print(challenge.islower())
```

```
False
```

```
In [ ]:
```

isnumeric(): check numeric char

```
In [4]: num='10'  
print(num.isnumeric())
```

```
True
```

```
In [6]: print('ten'.isnumeric())
```

```
False
```

join(): return a concatenated string

```
In [9]: web_tech=['HTML','CSS','javascript']  
result='#'.join(web_tech)  
print(result)
```

```
HTML#CSS#javascript
```

strip(): removes both leading and trailing characters

```
In [10]: challenge='thirty days of python'  
print(challenge.strip('y'))
```

```
thirty days of python
```

```
In [ ]:
```

replace(): Replace substring inside

```
In [13]: challenge = ' thirty days of python '  
print(challenge.replace('python','coding'))
```

```
thirty days of coding
```

```
In [ ]:
```

split(): Splits string from left

```
In [15]: challenge = ' thirty days of python '  
print(challenge.split())
```

```
['thirty', 'days', 'of', 'python']
```

```
In [ ]:
```

title(): Return a title cased string

```
In [16]: challenge = ' thirty days of python '  
print(challenge.title())
```

```
Thirty Days Of Python
```

```
In [ ]:
```

swapcase(): checks if string starts with the specified string

```
In [17]: challenge = ' thirty days of python '  
print(challenge.swapcase())
```

```
THIRTY DAYS OF PYTHON
```

```
In [ ]:
```

```
In [19]: challenge = ' Thirty Days Of Python '
print(challenge.swapcase())
```

```
tHIRTY dAYS oF pYTHON
```

```
In [ ]:
```

startswith(): checks if string starts with the specified string

```
In [21]: challenge='thirty days of python'
print(challenge.startswith('thirty'))
```

```
True
```

```
In [ ]:
```

```
In [22]: challenge='30 days of python'
print(challenge.startswith('thirty'))
```

```
False
```

```
In [ ]:
```