

```
In [5]: greeting='hello'  
greeting
```

```
Out[5]: 'hello'
```

```
In [6]: len(greeting)  
len('hello,world')
```

```
Out[6]: 11
```

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

```
Out[7]: 5
```

```
In [9]: space=' '  
space
```

```
Out[9]: ' '
```

string concatenation

```
In [14]: first_name='Rajesh'  
last_name='Jangam'  
full_name=first_name + ' '+last_name  
print(full_name)
```

```
Rajesh Jangam
```

```
In [15]: print(len(first_name) > len(last_name))
```

```
False
```

Unpacking characters

```
In [16]: language='Python'  
a,b,c,d,e,f=language    #unpacking sequence characters into variables  
print(a)  
print(b)  
print(c)  
print(d)  
print(e)  
print(f)
```

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

Accessing the characters in strings by index

```
In [18]: first_letter=language[0]
         print(first_letter)
```

P

```
In [19]: last_index=len(language)-1
         last_letter=language[last_index]
         print(last_letter)
```

n

```
In [21]: last=language[-3]
         print(last)
```

h

Slicing

```
In [22]: last_four=language[2:6]
         print(last_four)
```

thon

```
In [23]: last_three=language[-2:]  #another way
         print(last_three)
```

on

```
In [24]: language
```

```
Out[24]: 'Python'
```

```
In [25]: pto=language[0:6:2]  # skipping character while splitting python strings
         print(pto)
```

Pto

```
In [30]: 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
-------	---	---

This is a back slash symbol (\\)

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

```
In [27]: print('Days\tTopics\tExercises')
```

Days	Topics	Exercises
------	--------	-----------

```
In [28]: print('Day 1\t3\t5')
```

Day 1 3 5

String Methods

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

```
In [31]: challenge = 'thirty days of python'
print(challenge.capitalize()) # 'Thirty days of python'
```

Thirty days of python

count(): returns occurrences of substring in string, count(substring, start=.., end=..)

```
In [39]: challenge = 'thirty days of python '
print(challenge.count('y')) # 3
print(challenge.count('y', 7, 14)) # 1
print(challenge.count('th')) # 2`
```

3
1
2

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

```
In [41]: challenge = 'thirty days of python'
print(challenge.endswith('on')) # True
print(challenge.endswith('tion')) # False
```

True
False

expandtabs(): Replaces tab character with spaces, default tab size is 8. It takes tab size argument

```
In [42]: challenge = 'thirty\tdays\tof\tpython'
print(challenge.expandtabs())
print(challenge.expandtabs(10))
```

thirty days of python
thirty days of python

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

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

5
0

format() formats string into nicer output

```
In [44]: first_name = 'Asabeneh'
last_name = 'Yetayeh'
job = 'teacher'
country = 'Finland'
sentence = 'I am {} {}. I am a {}. I live in {}.'.format(first_name, last_name,
print(sentence)
```

I am Asabeneh Yetayeh. I am a teacher. I live in Finland.

```
In [45]: 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 3.14

index(): Returns the index of substring

```
In [46]: challenge = 'thirty days of python'
print(challenge.find('y'))
print(challenge.find('th'))
```

5
0

isalnum(): Checks alphanumeric character

```
In [47]: challenge = 'ThirtyDaysPython'
print(challenge.isalnum()) # True

challenge = '30DaysPython'
print(challenge.isalnum()) # True

challenge = 'thirty days of python'
print(challenge.isalnum()) # False

challenge = 'thirty days of python 2019'
print(challenge.isalnum()) # False
```

True
True
False
False

isalpha(): Checks if all characters are alphabets

```
In [53]: challenge = 'thirty days of python'
print(challenge.isalpha()) # True
num = '123'
print(num.isalpha())      # False
a='abcdef'
print(a.isalpha())
a='abfdcdf'
print(a.isalpha())
a='abfd cdef'
print(a.isalpha())
```

False
False
True
True
False

isdecimal(): Checks Decimal Characters

```
In [54]: challenge = 'thirty days of python'
print(challenge.find('y')) # 5
print(challenge.find('th')) # 0
```

5
0

isdigit(): Checks Digit Characters

```
In [56]: challenge = 'Thirty'
print(challenge.isdigit())
```

False

isdigit(): Checks Digit Characters

```
In [58]: challenge = '30'
print(challenge.isdigit())
```

True

```
In [59]: # isdecimal():Checks decimal characters

num = '10'
```

```
print(num.isdecimal())
num = '10.5'
print(num.isdecimal())
```

True
False

In [60]: *# isidentifier():Checks for valid identifier means it check if a string is a val*

```
challenge = '30DaysOfPython'
print(challenge.isidentifier()) # False, because it starts with a number
challenge = 'thirty_days_of_python'
print(challenge.isidentifier()) # True
```

False
True

In [61]: *# islower():Checks if all alphabets in a string are lowercase*

```
challenge = 'thirty days of python'
print(challenge.islower()) # True
challenge = 'Thirty days of python'
print(challenge.islower()) # False
```

True
False

In [62]: *# isupper(): returns if all characters are uppercase characters*

```
challenge = 'thirty days of python'
print(challenge.isupper()) # False
challenge = 'THIRTY DAYS OF PYTHON'
print(challenge.isupper()) # True
```

False
True

In [63]: *# isnumeric():Checks numeric characters*

```
num = '10'
print(num.isnumeric()) # True
print('ten'.isnumeric()) # False
```

True
False

In [64]: *# join(): Returns a concatenated string*

```
web_tech = ['HTML', 'CSS', 'JavaScript', 'React']
result = '#, '.join(web_tech)
print(result) # 'HTML# CSS# JavaScript# React'
```

HTML#, CSS#, JavaScript#, React

In [66]: *# strip(): Removes both leading and trailing characters*

```
challenge = ' thirty days of python '
print(challenge.strip('y')) # 5
```

thirty days of python

In [67]: *# replace(): Replaces substring inside*

```
challenge = 'thirty days of python'  
print(challenge.replace('python', 'coding')) # 'thirty days of coding'
```

thirty days of coding

In [68]: *# split(): Splits String from Left*

```
challenge = 'thirty days of python'  
print(challenge.split()) # ['thirty', 'days', 'of', 'python']
```

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

In [69]: *# title(): Returns a Title Cased String*

```
challenge = 'thirty days of python'  
print(challenge.title()) # Thirty Days Of Python
```

Thirty Days Of Python

In [70]: *# swapcase(): Checks if String Starts with the Specified String*

```
challenge = 'thirty days of python'  
print(challenge.swapcase()) # THIRTY DAYS OF PYTHON  
challenge = 'Thirty Days Of Python'  
print(challenge.swapcase()) # tHIRTY dAYS oF pYTHON
```

THIRTY DAYS OF PYTHON

tHIRTY dAYS oF pYTHON

In [71]: *# startswith(): Checks if String Starts with the Specified String*

```
challenge = 'thirty days of python'  
print(challenge.startswith('thirty')) # True  
challenge = '30 days of python'  
print(challenge.startswith('thirty')) # False
```

True

False

In []: