

# Day 04 – Strings and String Methods in Python (Extended)

---

Practice advanced string manipulations and explore a wide range of Python string methods.

## String Declaration, Length, and Printing

```
In [1]: letter = 'P'
print(letter)
print(len(letter))

greeting = 'Hello, World!'
print(greeting)
print(len(greeting))

sentence = "I hope you are enjoying 30 days of python challenge"
print(sentence)
```

```
P
1
Hello, World!
13
I hope you are enjoying 30 days of python challenge
```

## Multiline Strings

```
In [7]: 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)

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.
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.
```

## String Concatenation and Length

```
In [8]: first_name = 'Arman'
last_name = 'Pathan'
space = ' '
full_name = first_name + space + last_name
print(full_name)
print(len(first_name))
print(len(last_name))
print(len(full_name))
```

```
Arman Pathan
5
6
12
```

## Unpacking and Indexing

```
In [9]: language = 'Python'
a, b, c, d, e, f = language
print(a, b, c, d, e, f)
```

P y t h o n

```
In [10]: first_letter = language[0]
last_letter = language[-1]
second_last = language[-2]
print(first_letter, last_letter, second_last)
```

P n o

## Slicing and Skipping

```
In [11]: print(language[0:3])
print(language[3:6])
print(language[-3:])
print(language[::-2])
```

Pyt  
hon  
hon  
Pto

## Escape Sequences

```
In [13]: # Escape sequence
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!"

## String Methods

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

```
In [28]: 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 [29]: 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 [30]: 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 [31]: challenge = 'thirty\tdays\tof\tpython'
print(challenge.expandtabs()) # 'thirty  days    of      python'
print(challenge.expandtabs(10)) # 'thirty  days    of      python'
```

thirty days of python  
thirty days of python

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

```
In [32]: challenge = 'thirty days of python'
print(challenge.find('y')) # 5
print(challenge.find('th')) # 0
```

50

## format() formats string into nicer output

```
In [33]: first_name = 'Arman'
last_name = 'Pathan'
job = 'Ai Engineer'
country = 'India'
sentence = 'I am {} {}. I am an {}. I live in {}'.format(first_name, last_name, job, country)
print(sentence)
```

I am Arman Pathan. I am an Ai Engineer. I live in India.

```
In [34]: 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
```

The area of circle with 10 is 3.14

**index():** Returns the index of substring

```
In [35]: challenge = 'thirty days of python'
print(challenge.find('y')) # 5
print(challenge.find('th')) # 0
```

50

**isalnum():** Checks alphanumeric character

```
In [36]: 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 [37]: challenge = 'thirty days of python'
print(challenge.isalpha()) # True
num = '123'
print(num.isalpha())      # False
```

False  
False

## isdecimal(): Checks Decimal Characters

```
In [38]: challenge = 'thirty days of python'
print(challenge.find('y')) # 5
print(challenge.find('th')) # 0
```

5  
0

## isdigit(): Checks Digit Characters

```
In [41]: challenge = 'Thirty'
print(challenge.isdigit()) # False
challenge = '30'
print(challenge.isdigit()) # True
```

False  
True

## isdecimal():Checks decimal characters

```
In [42]: num = '10'
print(num.isdecimal()) # True
num = '10.5'
print(num.isdecimal()) # False
```

True  
False

## isidentifier():Checks for valid identifier means it check if a string is a valid variable name

```
In [43]: challenge = '30DaysOfPython'
print(challenge.isidentifier()) # False, because it starts with a number
challenge = 'thirty_days_of_python'
print(challenge.isidentifier()) # True
```

False  
True

## islower():Checks if all alphabets in a string are lowercase

```
In [44]: challenge = 'thirty days of python'
print(challenge.islower()) # True
challenge = 'Thirty days of python'
print(challenge.islower()) # False
```

True  
False

**isupper():** returns if all characters are uppercase characters

```
In [45]: challenge = 'thirty days of python'
print(challenge.isupper()) # False
challenge = 'THIRTY DAYS OF PYTHON'
print(challenge.isupper()) # True
```

False  
True

**isnumeric():** Checks numeric characters

```
In [46]: num = '10'
print(num.isnumeric()) # True
print('ten'.isnumeric()) # False
```

True  
False

**join():** Returns a concatenated string

```
In [47]: web_tech = ['HTML', 'CSS', 'JavaScript', 'React']
result = '#'.join(web_tech)
print(result) # 'HTML# CSS# JavaScript# React'
```

HTML#, CSS#, JavaScript#, React

**strip():** Removes both leading and trailing characters

```
In [48]: challenge = ' thirty days of python '
print(challenge.strip('y')) # 5
```

thirty days of python

**replace():** Replaces substring inside

```
In [49]: challenge = 'thirty days of python'
print(challenge.replace('python', 'coding')) # 'thirty days of coding'
```

thirty days of coding

**split():** Splits String from Left

```
In [50]: challenge = 'thirty days of python'
print(challenge.split()) # ['thirty', 'days', 'of', 'python']
```

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

**title():** Returns a Title Cased String

```
In [51]: challenge = 'thirty days of python'
print(challenge.title()) # Thirty Days Of Python
```

Thirty Days Of Python

**swapcase():** Checks if String Starts with the Specified String

```
In [52]: 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

## startswith(): Checks if String Starts with the Specified String

```
In [53]: challenge = 'thirty days of python'
print(challenge.startswith('thirty')) # True
challenge = '30 days of python'
print(challenge.startswith('thirty')) # False
```

True

False