

Python - Strings

In [1]:

```
1 """This is a
2 docstring"""
```

Out[1]:

```
'This is a\ndocstring'
```

In [2]:

```
1 print('This is a \ndocstring')
```

```
This is a
docstring
```

In [2]:

```
1 string = 'Words in Single / Double Quotes.'
```

Data Type - String

In [4]:

```
1 print(type('string'))
```

```
<class 'str'>
```

Functions - Strings

In [7]:

```
1strf = ['capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expand
2      'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isd
3      'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust
4      'maketrans', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartit
5      'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'u
```

capitalize - Converts your first letter of the Sentence in Upper Case.

In [8]:

```
1 ph = 'machine Learning is a subset of Artificial Intelligence.'
```

In [9]:

```
1 print(ph)
```

```
machine Learning is a subset of Artificial Intelligence.
```

In [11]:

```
1 print(ph.capitalize())
```

Machine learning is a subset of artificial intelligence.

casefold : returns you a compared versions of your string

In [12]:

```
1 help('str.casefold')
```

Help on method_descriptor in str:

```
str.casefold = casefold(self, /)
    Return a version of the string suitable for caseless comparisons.
```

In [13]:

```
1 ph.casefold()
```

Out[13]:

```
'machine learning is a subset of artificial intelligence.'
```

center : marks your string to the center space you give

In [15]:

```
1 'word'.center(50)
```

Out[15]:

```
'                               word                               '
```

count : return you the iterations of your particular test

In [21]:

```
1 print(ph)
2 print(ph.casefold().count('l'))
```

machine Learning is a subset of Artificial Intelligence.

4

index : Returns you the index positions

In [24]:

```
1 ph.lower().index('l'), ph.index('l')
```

Out[24]:

```
(8, 41)
```

In [25]:

```
1 ph[8], ph[41]
```

Out[25]:

```
('L', 'l')
```

find : this helps to again identify the index positions

In [26]:

```
1 ph.find('l')
```

Out[26]:

```
41
```

Find vs Index

In [27]:

```
1 'word'.find('s')
```

Out[27]:

```
-1
```

In [28]:

```
1 'word'.index('s')
```

```
-----  
-----  
ValueError                                Traceback (most recent call  
  last)  
<ipython-input-28-121bc177dd18> in <module>  
----> 1 'word'.index('s')
```

ValueError: substring not found

Format : this returns a formatted version of the string within the {}

In [29]:

```
1 df = "{} is a function of {}"
```

In [31]:

```
1 df
```

Out[31]:

```
'{} is a function of {}'
```

In [32]:

```
1 print(df.format('DataFrame', 'Pandas'))
```

DataFrame is a function of Pandas

In [34]:

```
1 "{} + {} = {}".format(2, 3, 5)
```

Out[34]:

'2 + 3 = 5'

isalpha

In [35]:

```
1 'hrevfuidke'.isalpha()
```

Out[35]:

True

In [37]:

```
1 '1257632hjd'.isalpha()
```

Out[37]:

False

isalnum

In [39]:

```
1 'c3eni43'.isalnum()
```

Out[39]:

True

join : joins the strings together

In [40]:

```
1 help('str.join')
```

Help on method_descriptor in str:

```
str.join = join(self, iterable, /)
Concatenate any number of strings.
```

The string whose method is called is inserted in between each given string.

The result is returned as a new string.

Example: `'.'.join(['ab', 'pq', 'rs']) -> 'ab.pq.rs'`

In [41]:

```
1 ".".join(['www', 'edufabrica', 'net'])
```

Out[41]:

```
'www.edufabrica.net'
```

In [42]:

```
1 user = input('enter email : ')
```

```
enter email : iitd.shukla
```

In [43]:

```
1 usl = [user, 'gmail.com']
```

In [44]:

```
1 email_user = "@".join(usl)
```

In [45]:

```
1 email_user
```

Out[45]:

```
'iitd.shukla@gmail.com'
```

lower : converts the string to lower case

In [46]:

```
1 'WORK'.lower()
```

Out[46]:

```
'work'
```

upper : converts complete string to upper case

In [47]:

```
1 'work'.upper()
```

Out[47]:

```
'WORK'
```

title : converts your first letter of every occuring word in the string to upper case

In [48]:

```
1 df = 'hello, my name is sumit!'
```

In [49]:

```
1 df.title()
```

Out[49]:

```
'Hello, My Name Is Sumit!'
```

replace : replace the substrings given

In [50]:

```
1 df
```

Out[50]:

```
'hello, my name is sumit!'
```

In [53]:

```
1 df = df.replace('h','H')
```

In [54]:

```
1 df
```

Out[54]:

```
'Hello, my name is sumit!'
```

swapcase : cases will be swapped accordingly to lower / upper

In [55]:

```
1 qwer = 'QweRtYuIKhgh'
```

In [56]:

```
1 qwer, qwer.swapcase()
```

Out[56]:

```
('QweRtYuIKhgh', 'qWErTyUikHGH')
```

In [60]:

```
1 df.replace('m','M',1)
```

Out[60]:

```
'Hello, My name is sumit!'
```

Task

In [61]:

```
1 a = 100
2 b = 80
3 c = 90
```

In [62]:

```
1 1008090
```

Out[62]:

1008090

In [63]:

```
1 print(str(a)+ str(b)+ str(c))
```

1008090

In [66]:

```
1 print("100"+"80"+"90")
```

1008090

In []:

```
1
```