

Topics

- Methods in strings
- some tasks on strings
- Data Structures

```
In [4]: 1 s="Python Programming"
        2 type(s)
```

Out[4]: str

```
In [3]: 1 s[7:20]
```

Out[3]: 'Programming'

```
In [5]: 1 print(dir(str))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattr__', '__getitem__', '__getnewa
rgs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__l
e__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce
__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__siz
eof__', '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center', 'cou
nt', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'inde
x', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'i
slower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join',
'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rind
ex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startsw
ith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

```
In [37]: 1 s = "python programming"
        2 print(s.capitalize())
        3 s = "PythoN programming"
        4 print(s.capitalize())
        5
```

Python programming
Python programming

```
In [11]: 1 s = "python programming"
        2 s.istitle()
        3 s.title()
```

Out[11]: 'Python Programming'

```
In [12]: 1 s = "PythoN programming"
2 print(s.casefold())
3 print(s.islower())
4 print(s.lower())
5 print(s.isupper())
6 print(s.upper())
```

```
python programming
False
python programming
False
PYTHON PROGRAMMING
```

```
In [14]: 1 s = "Python"
2 print(s.isalpha())
3 s1 = "Python Programming" # pythonprogramming
4 print(s1.isalpha())
```

```
True
False
```

```
In [20]: 1 s1 = "Python Programming"
2
3 for i in s1:
4     print(i,end= " ")
5 print()
6 for i in range(len(s1)): # for i in range(18)
7     print(s1[i], end = ' ')
8
```

```
P y t h o n   P r o g r a m m i n g
P y t h o n   P r o g r a m m i n g
```

```
In [23]: 1 s1 = "Python Programming"
2 s2 = ""
3 for i in s1:
4     if i.isalpha():
5         s2+=i
6 print(s2)
7
```

```
PythonProgramming
```

$U_{\sim ymts}(\cdot)$

```
1 ord("A")
```

```
1 ord("A")
```

```
1 chr(65)
```

```
1 chr(65)
```

```
1 s = "Python Programmming Python"
2 s.count("Python")
```

```
1 s = "Python Programmming Python"
2 s.count("Python")
```

```
1 s[5]
```

```
1 s[5]
```

```
1 s.index("g")
```

```
1 s.index("g")
```

```
1 s= "Python Programming Python"
2 for i in range(len(s)):
3     if s[i] == "g":
4         print(i)
```

```
1 s= "Python Programming Python"
2 for i in range(len(s)):
3     if s[i] == "g":
4         print(i)
```

```
1 s = "12344 123"
2 s.isdigit()
```

```
1 s = "12344 123"
2 s.isdigit()
```

```
In [51]: 1 s = "Python123"
         2 print(s.isalnum())
         3 s1 = "Python 123"
         4 print(s1.isalnum())
         5
```

True
False

```
In [56]: 1 s= "Python Programmming Python"
         2 s1 = s.split()
         3 type(s1)
```

Out[56]: list

```
In [59]: 1 s = "Python 123".split(" ")
         2 type(s)
         3 s
```

Out[59]: ['Python', '123']

```
In [62]: 1 s = "Python123"
         2 s.split("@")
```

Out[62]: ['Python123']

```
In [64]: 1 "$".join(s)
```

Out[64]: 'P\$y\$t\$h\$o\$n\$1\$2\$3'

```
In [69]: 1 s = "Python programming".split()
         2 print(s)
         3 "$".join(s)
```

['Python', 'programming']

Out[69]: 'Python\$programming'

```
In [72]: 1 s ="Python\nProgramming\n123"
         2 s.splitlines()
         3 s.split("\n")
```

Out[72]: ['Python', 'Programming', '123']

```
In [79]: 1 s = "Python programming" # Python$programming
         2 s.replace(" ", "$")
         3 s.replace(" ", "")
```

Out[79]: 'Pythonprogramming'

```
In [77]: 1 s1 = "          Python          "
2 print(s.lstrip())
3 print(s.rstrip())
4 print(s.strip())
```

Python

Python

Python

```
In [84]: 1 s.center(21,"@")
```

Out[84]: '@@Python programming@'

```
In [82]: 1 len(s)
```

Out[82]: 18

```
1 1. s = "Python Programming"
2
3 output: PYTHON programming
4
5
6 2. s = "Pyt@hon P#rogra$mming"
7
8 output: @# $
```

```
In [89]: 1 s = "Python Programming"
2 t = s.split()
3 print(t)
4 final = t[0].upper()
5 final+=" "+t[1]
6
```

['Python', 'Programming']

Out[89]: 'PYTHON Programming'

```
In [90]: 1 s = "Pyt@hon P#rogra$mming"
2 s1 = ""
3 for i in s:
4     if i.isalpha()or i.isdigit():
5         continue
6     else:
7         s1+=i
8 print(s1)
```

@ # \$

```
In [ ]: 1
```

