

```
In [ ]: Continuation of strings:
```

```
In [6]: full_name = 'amritha vijayan'
print(full_name)
```

amritha vijayan

```
In [2]: type(full_name)
```

Out[2]: str

```
In [7]: #to print first letter of word in capitals
print(full_name.title())
```

Amritha Vijayan

```
In [ ]: title() ----->convert the first letter of word to capital letter
```

```
In [8]: #print entire name in capitals
print(full_name.upper())
```

AMRITHA VIJAYAN

```
In [9]: #print name in small letters
print(full_name.lower())
```

amritha vijayan

```
In [ ]: Introduction to f strings:
```

```
In [ ]: first_name = 'amritha'
last_name = 'vijayan'
```

```
In [ ]: # requirement is to print full name
```

```
In [ ]: #syntax of f strings:
f" custom words {placeholder1} {placeholder2}..... ..{placeholdern}"
```

```
In [23]: full_name = f"{first_name} {last_name}"
print(full_name)
```

amritha vijayan

```
In [24]: print(full_name.title())
```

Amritha Vijayan

```
In [25]: print(full_name.upper())
```

AMRITHA VIJAYAN

```
In [27]: print(f"Hello, {full_name.title()}")
```

Hello, Amritha Vijayan

```
In [28]: #Adding white spaces to strings(\n and \t):  
print("Fav_Prog_Language:PythonC++CobalPascalJava")
```

Fav\_Prog\_Language:PythonC++CobalPascalJava

```
In [29]: print("Fav_Prog_Language:\nPython\nC++\nCobal\nPascal\nJava")
```

Fav\_Prog\_Language:  
Python  
C++  
Cobal  
Pascal  
Java

```
In [ ]: \n is a new line delimiter  
       \t is tab delimiter
```

```
In [30]: print("Fav_Prog_Language:\n\tPython\n\tC++\n\tCobal\n\tPascal\n\tJava")
```

Fav\_Prog\_Language:  
 Python  
 C++  
 Cobal  
 Pascal  
 Java

```
In [35]: #Removing white spaces from the strings:  
language1 = 'Python'  
print(language1)
```

Python

```
In [38]: language2 = ' Java'  
print(language2)
```

Java

```
In [37]: language3 = 'C++ '  
print(language3)
```

C++

```
In [ ]: these whitespaces will consume the storage and needs to be removed
```

```
In [39]: language2.lstrip() #left side stripping
```

Out[39]: 'Java'

```
In [41]: language3.rstrip() #right side stripping
```

Out[41]: 'C++'

```
In [42]: name = ' amritha '  
print(name)
```

```
amritha
```

```
In [43]: name.strip() #both side stripping
```

```
Out[43]: 'amritha'
```

```
In [ ]: why strip() is not used for above cases lstrip() and rstrip()?==>this consumes time during execution, it could be  
on production environment.  
2 steps involved-->search and then eliminating space  
if l and r is used it straight away eliminate the respective side
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js