

String slicing

String[start: end: step]

```
In [1]: name='yasaswini'  
print(name[1])
```

a

```
In [2]: color='skyblue'  
print(color[0:1]) #1st char
```

s

```
In [3]: print(color[:1])#1st char
```

s

```
In [4]: print(color[0:4])
```

skyb

```
In [5]: print(color[:4]) #color[0:4]=color[:4]=skyb
```

skyb

```
In [6]: print(color[3:])
```

blue

```
In [7]: print(color[-4:]) #color[3:]=color[-4:]=blue
```

blue

```
In [8]: print(color[2:6])
```

yblu

```
In [9]: #last char  
print(color[-1:])
```

e

```
In [10]: print(color[6:]) #last char
```

e

```
In [11]: #Last 3 chars  
print(color[4:])
```

lue

```
In [12]: print(color[-3:]) #Last 3 chars
```

lue

```
In [13]: #every second char  
print(name[0::2])
```

yssii

```
In [15]: print(name[0:9:2]) #every second char
```

yssii

```
In [16]: # every 3 char  
print(name[0::3])
```

yai

```
In [17]: #exclude first and last char  
print(color[1:-1])
```

kyblu

```
In [18]: #print all chars  
print(name[:])
```

yasaswini

```
In [19]: print(color[:]) #print all chars
```

skyblue

```
In [20]: print(name[::-1]) #reverse the string
```

iniwsasay

```
In [21]: a='Slicing in Python is a feature that enables accessing parts of the sequence  
print(a[0:9])
```

Slicing i

```
In [22]: print(a[0:-1])
```

Slicing in Python is a feature that enables accessing parts of the sequenc

```
In [23]: #reverse the string
print(a[::-1])
```

ecneuqes eht fo strap gnissec ca selbane taht erutaef a si nohtyP ni gnicils

```
In [24]: print(a[-12:])
```

the sequence

```
In [25]: print(a[-30:])
```

ccessing parts of the sequence

```
In [26]: #exclude first and last chars
print(a[1:-1])
```

licing in Python is a feature that enables accessing parts of the sequenc

String methods

len()-returns the length of string(number of characters)

```
In [27]: a='Slicing in Python is a feature that enables accessing parts of the sequence'
print(len(a))
```

75

upper() -converts a string into uppercase

```
In [28]: print(a.upper())
```

SLICING IN PYTHON IS A FEATURE THAT ENABLES ACCESSING PARTS OF THE SEQUENCE

lower() -converts a string into lowercase

```
In [29]: print(a.lower())
```

slicing in python is a feature that enables accessing parts of the sequence

strip() -remove any spaces,tabs or newline characters

```
In [30]: print(a.strip()) #it doesnt remove spaces in middle of string ,removes unwanted
```

Slicing in Python is a feature that enables accessing parts of the sequence

count() -returns a number of times a specified value occurs in a string

```
In [32]: print(a.count('a'))
```

```
6
```

find() -searches the string for a specified value and returns the position of where it was found

```
In [34]: print(a.find('a'))
```

```
21
```

title() - convert the first character of each word to uppercase

```
In [35]: print(a.title())
```

```
Slicing In Python Is A Feature That Enables Accessing Parts Of The Sequence
```

split()- split the string at specified separator and returns a list

```
In [39]: print(a.split())
```

```
['Slicing', 'in', 'Python', 'is', 'a', 'feature', 'that', 'enables', 'accessing', 'parts', 'of', 'the', 'sequence']
```

replace(old,new)- replace all the occurrences of a substring with a new string

```
In [40]: print(a.replace('a', 'an'))
```

```
Slicing in Python is an feature that enables accessing parts of the sequence
```

Break-used to stop the loop when a condition is met

```
In [42]: for i in range(1,11):
    if i==5:
        break
    print(i)
```

```
1
2
3
4
```

Continue -skips the current iteration and move to next

```
In [43]: for i in range(1,11):
    if i==5:
        continue
    print(i)
```

```
1
2
3
4
6
7
8
9
10
```

Pass -Does nothing but keeps code syntactically correct

```
In [44]: for i in range(1,6):
    if i==3:
        pass
    print(i)
```

```
1
2
3
4
5
```

```
In [ ]:
```