Python Fundamentals
day 16

Today's Agenda

- String slicing
- Reversing a string
- Programs on string

String slicing

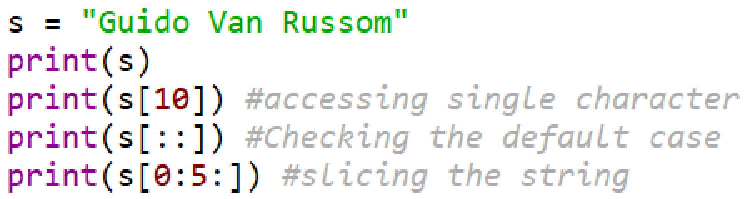
We have seen how to access single characters from a string. Now let us see how to take portion of a string, taking a portion of string is technically called as slicing.

Considering the string Guido Van Russom let us see how slicing can be done

Syntax: inclusive
↑
string_name [start : stop+1 : step]
↓
exclusive

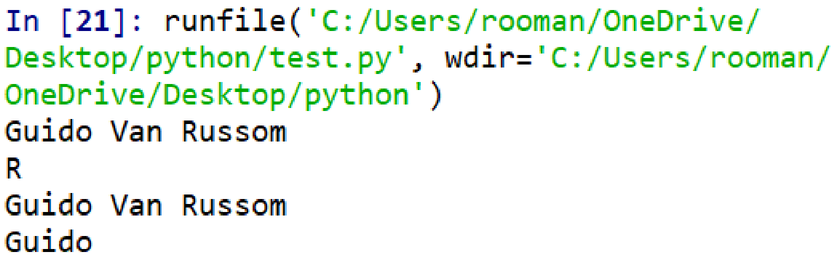
You can return a range of characters by using the slice syntax. Specify the start index and the end index, separated by a colon, to return a part of the string.

If not mentioned anything then by default start value is 0, stop value is the length of string + 1 and the step is 1.



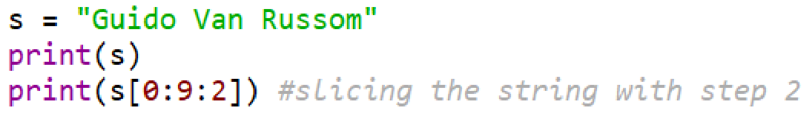
```
s = "Guido Van Russom"
print(s)
print(s[10]) #accessing single character
print(s[::]) #Checking the default case
print(s[0:5:]) #slicing the string
```

Output:



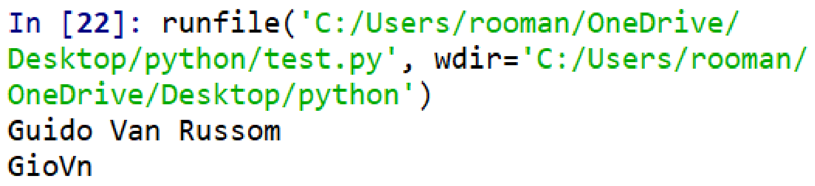
```
In [21]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')
Guido Van Russom
R
Guido Van Russom
Guido
```

We have seen how to access a single character, a slice of string. Let us now see how step value works



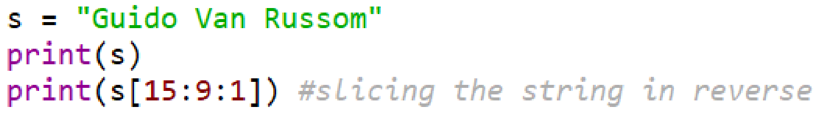
```
s = "Guido Van Russom"
print(s)
print(s[0:9:2]) #slicing the string with step 2
```

Output:



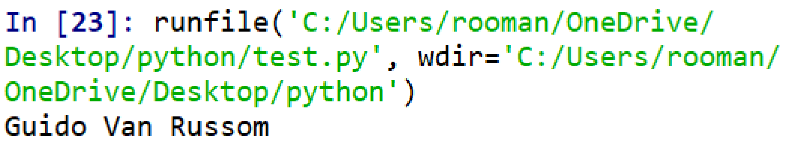
```
In [22]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')
Guido Van Russom
GioVn
```

Now let us try slicing the string in reverse way



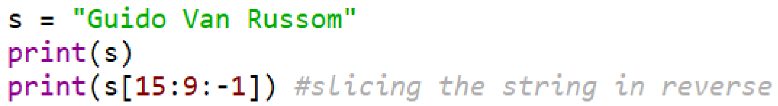
```
s = "Guido Van Russom"
print(s)
print(s[15:9:1]) #slicing the string in reverse
```

Output:



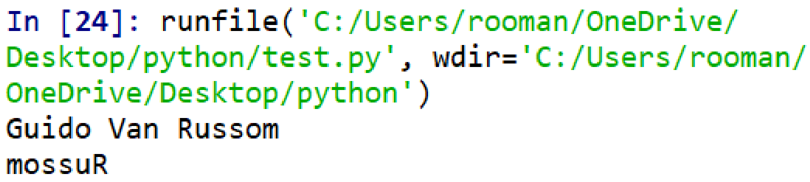
```
In [23]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')
Guido Van Russom
```

The reason we did not get the string in reverse is because, the step value is 1 which represents forward direction. If we want in reverse direction then step value should be negative.



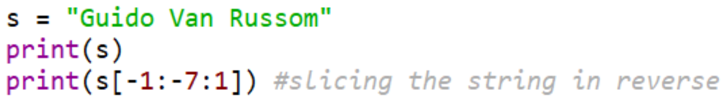
```
s = "Guido Van Russom"
print(s)
print(s[15:9:-1]) #slicing the string in reverse
```

Output:



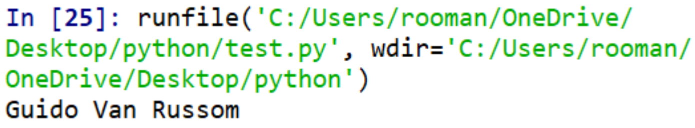
```
In [24]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')
Guido Van Russom
mossuR
```

In python we also have an additional feature of negative indexes which is much easier while printing a string in reverse.



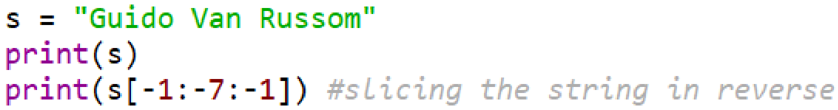
```
s = "Guido Van Russom"
print(s)
print(s[-1:-7:1]) #slicing the string in reverse
```

Output:



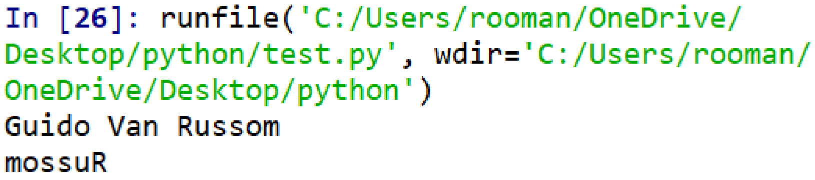
```
In [25]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')
Guido Van Russom
```

Irrespective of what the start and stop values are, if the step is 1 it always represents forward direction.



```
s = "Guido Van Russom"
print(s)
print(s[-1:-7:-1]) #slicing the string in reverse
```

Output:



```
In [26]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')
Guido Van Russom
mossuR
```

Reversing a string

Unlike other programming languages where reversing a string takes few set of lines, in python by knowing all the above techniques we can easily reverse a string.

Let us see how to do it

```
s = "Guido Van Russom"
print(s)
print(s[::-1]) #reverse of string
```

Output:

```
In [27]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')
Guido Van Russom
mossuR naV odiuG
```

Note: If we don't mention start and stop values and assign step value as -1. The starting value will be -1 and ending value or stop value will be negative of length of string.

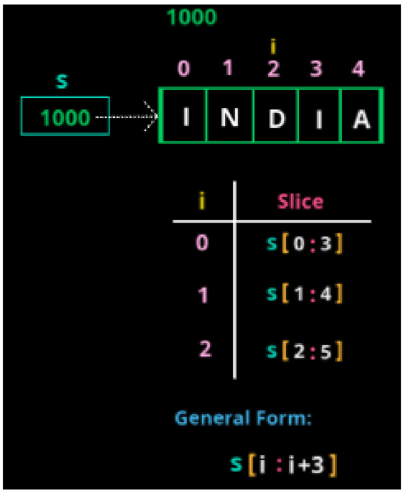
Programs on strings

1. **Program to print all the substrings with length 3 from the given string**

```
s=input("Enter the string:\n")

for i in range(0,len(s)-2):
    print(s[i:i+3])
```

Output:

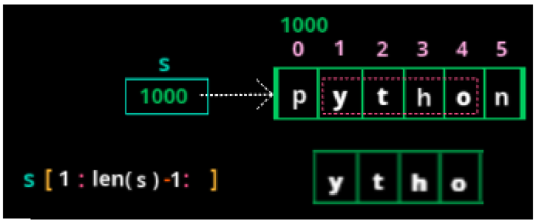
```
In [29]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')

Enter the string:
india
ind
ndi
dia
```



You can try the same program with as many strings as you want.

2. Program to print all the characters of a string except the first and the last character.

```
s=input("Enter the string:\n")
print(s[1:len(s)-1:])
```

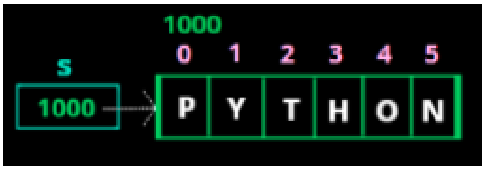


Output:

```
In [31]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')

Enter the string:
python
ytho
```

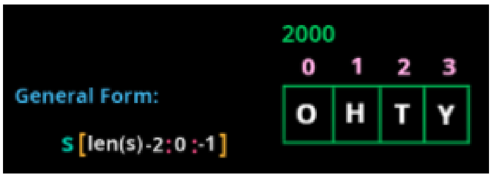
3. Program to print the characters of a string in the reverse order except the first and the last character

```
s=input("Enter the string:\n")
print(s[len(s)-2:0:-1])
```



Output:

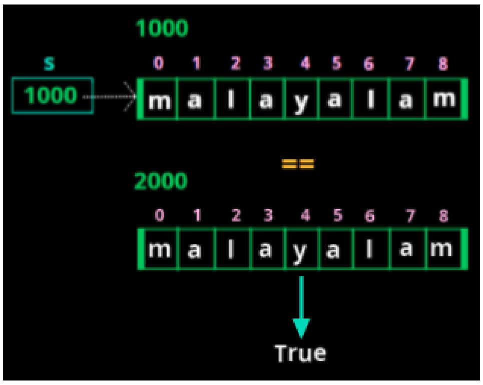
```
In [32]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')

Enter the string:
python
ohty
```

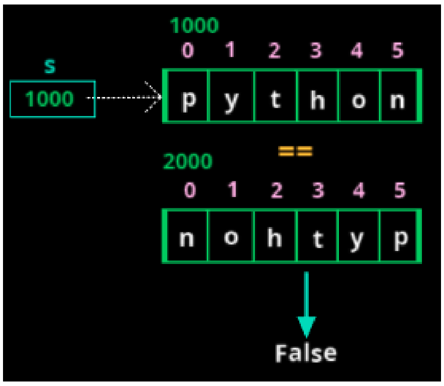


4. Program to check whether the given string is palindrome

```
s=input("Enter the string:\n")

if s==s[::-1]:
    print(s," String is palindrome")
else:
    print(s," String is not a palindrome")
```

Let us try the above with two example : Malayalam, python and see the output





Output:

```
In [33]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')

Enter the string:
malayalam
malayalam  String is palindrome

In [34]: runfile('C:/Users/rooman/OneDrive/
Desktop/python/test.py', wdir='C:/Users/rooman/
OneDrive/Desktop/python')

Enter the string:
python
python  String is not a palindrome
```