How to reverse a string in Python?

The collection of Unicode characters is Python String. Python has various capabilities for string control, yet Python string library doesn't uphold the in-constructed "switch()" capability. However, there are numerous methods for reversing the string. The following reverse Python String method is being defined.

- **Using for loop**
- **Using while loop**
- **Using the slice operator**
- **Using the reversed() function**
- **Using the recursion**

Using for loop

Here, we will reverse the given string using for loop.

```
def reverse_string(str):
    str1 = ""   # Declaring empty string to store the reversed string
    for i in str:
        str1 = i + str1
    return str1    # It will return the reverse string to the caller function

str = "JavaTpoint"    # Given String
print("The original string is: ",str)
print("The reverse string is",reverse_string(str)) # Function call
```

Output:

```
('The original string is: ', 'JavaTpoint')
('The reverse string is', 'tniopTavaJ')
```

Explanation-

We have passed the str argument and declared the reverse_string() function in the code above. We have declared the empty string variable str1 in the body of the function, which will store the reversed string.

The for loop then iterated over each part of the given string, joining each character at the beginning and saving the results in the str1 variable.

After the total cycle, it returned the opposite request string str1 to the guest capability. The result was displayed on the screen.

Using while loop

We can also reverse a string using a while loop. Let's understand the following example.

Example -

```
# Reverse string
# Using a while loop

str = "JavaTpoint" #  string variable
print ("The original string  is : ",str)
reverse_String = ""  # Empty String
count = len(str) # Find length of a string and save in count variable
while count > 0:
    reverse_String += str[ count - 1 ] # save the value of str[count-1] in reverseString
    count = count - 1 # decrement index
print ("The reversed string using a while loop is : ",reverse_String)# reversed string
```

Output:

```
('The original string is : ', 'JavaTpoint')
('The reversed string using a while loop is : ', 'tniopTavaJ')
```

Explanation:

We have declared a str variable with a string value in the code above. We introduced some time circle with a worth of the string.

The value of str[count - 1] decreased the count value as it concatenated with the reverse_String during each iteration. Sometime finished its cycle and returned the opposite request string.

Using the slice ([]) operator

We can also reverse the given string using the **extended slice operator**. Let's understand the following example.

Example -

```
#  Reverse a string
# using  slice syntax
# reverse(str) Function to reverse a string
def reverse(str):
    str = str[::-1]
    return str

s = "JavaTpoint"
print ("The original string  is : ",s)
print ("The reversed string using extended slice operator  is : ",reverse(s))
```

Output:

```
('The original string is : ', 'JavaTpoint')
('The reversed string(using extended slice syntax) is : ', 'tniopTavaJ')
```

Explanation:

Start, stop, and step are the three parameters that a slice operator typically accepts. We offered the no benefit to begin and end file, which shows the beginning record is 0 and the end is n-1 of course. -1 is the step size; it implies the string proceeds with the navigate from the end and goes to the 1 file position.

Using reverse function with join

Python provides the **reversed()** function to reverse the string. Let's understand the following example.

Example -

```python
#reverse a string using reversed()
# Function to reverse a string
def reverse(str):
    string = "".join(reversed(str)) # reversed() function inside the join() function
    return string

s = "JavaTpoint"

print ("The original string is : ",s)
print ("The reversed string using reversed() is : ",reverse(s) )
```

Output:

```
('The original string is : ', 'JavaTpoint')
('The reversed string using reversed() is : ', 'tniopTavaJ')
```

Explanation:

We declared the empty string separated by the.dot operator in the body of the function. The reversed() string that it joined with the empty string separated by the join() function returned the reversed string.

Using recursion()

The recursion can also be used to turn the string around. Recursion is a cycle where capability calls itself. Look at the following example.

Example -

```python
# reverse a string
# using recursion

def reverse(str):
    if len(str) == 0: # Checking the lenght of string
        return str
    else:
        return reverse(str[1:]) + str[0]

str = "Devansh Sharma"
print ("The original string  is : ", str)
print ("The reversed string(using recursion) is : ", reverse(str))
```

Output:

```
('The original string is : ', 'JavaTpoint')
('The reversed string(using reversed) is : ', 'tniopTavaJ')
```

Explanation:

We have defined a function in the code above that takes the string as an argument.

In the capability body, we characterized the base state of recursion, in the event that a length of a string is 0, the string is returned, and while perhaps not then we called the capability recursively.

The first character of the string is concatenated to the end of the slice string by the slice operator.