

Python - caracteres de escape

Em Python, uma string se torna uma string bruta se for prefixada com "r" ou "R" antes dos símbolos de cotação. Portanto, 'Hello' é uma string normal, enquanto r'Hello' é uma string bruta.

```
>>> normal="Hello"
>>> print (normal)
Hello
>>> raw=r"Hello"
>>> print (raw)
Hello
```

Em circunstâncias normais, não há diferença entre os dois. No entanto, quando o caractere de escape está incorporado na string, a string normal realmente interpreta a sequência de escape, enquanto a string bruta não processa o caractere de escape.

```
>>> normal="Hello\nWorld"
>>> print (normal)
Hello
World
>>> raw=r"Hello\nWorld"
>>> print (raw)
Hello\nWorld
```

No exemplo acima, quando uma string normal é impressa, o caractere de escape '\n' é processado para introduzir uma nova linha. No entanto, devido ao operador de string bruto 'r', o efeito do caractere de escape não é traduzido de acordo com seu significado.

O caractere de nova linha \n é uma das sequências de escape identificadas pelo Python. A sequência de escape invoca uma subsequência de caracteres de implementação alternativa para "\". Em Python, "\"" é usado como caractere de escape. A tabela a seguir mostra uma lista de sequências de escape.

A menos que um prefixo 'r' ou 'R' esteja presente, as sequências de escape em literais de string e bytes são interpretadas de acordo com regras semelhantes àsquelas usadas pelo Padrão C. As sequências de escape reconhecidas são -

Sr. Não	Sequência e significado de escape
1	\<nova linha>

	Barra invertida e nova linha ignoradas
2	\\ Barra invertida (\)
3	\' Citação única (')
4	\" Citação dupla (")
5	\a Campainha ASCII (BEL)
6	\b Retrocesso ASCII (BS)
7	\f Alimentação de formulário ASCII (FF)
8	\n Alimentação de linha ASCII (LF)
9	\r Retorno de transporte ASCII (CR)
10	\t Guia Horizontal ASCII (TAB)
11	\v Guia Vertical ASCII (VT)
12	\ooo Caractere com valor octal ooo
13	\xhhh Caractere com valor hexadecimal hh

Exemplo

O código a seguir mostra o uso de sequências de escape listadas na tabela acima -

```
# ignore \
s = 'This string will not include \
backslashes or newline characters.'
print (s)

# escape backslash
```

```
s=s = 'The \\character is called backslash'
print (s)

# escape single quote
s='Hello \'Python\''
print (s)

# escape double quote
s="Hello \"Python\""
print (s)

# escape \b to generate ASCII backspace
s='Hel\blo'
print (s)

# ASCII Bell character
s='Hello\a'
print (s)

# newline
s='Hello\nPython'
print (s)

# Horizontal tab
s='Hello\tPython'
print (s)

# form feed
s= "hello\fworld"
print (s)

# Octal notation
s="\101"
print(s)

# Hexadecimal notation
s="\x41"
print (s)
```

Ele produzirá a seguinte **saída** -

This string will not include backslashes or newline characters.
The \character is called backslash
Hello 'Python'

Hello "Python"

Helo

Hello

Hello

Python

Hello Python

hello

world

A

A