Python Strings

Python String

- Python string is the collection of the characters surrounded by single quotes, double quotes, or triple quotes. The computer does not understand the characters; internally, it stores manipulated character as the combination of the 0's and 1's.
- Each character is encoded in the ASCII or Unicode character. So we can say that Python strings are also called the collection of Unicode characters.
- strings can be created by enclosing the character or the sequence of characters in the quotes. Python allows us to use single quotes, double quotes, or triple quotes to create the string

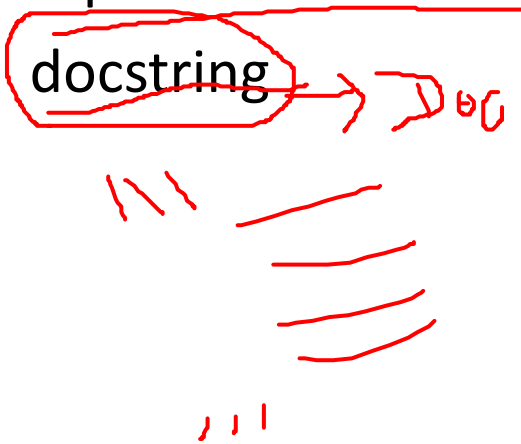
str = "Hi Python !"

print(type(str)), then it will **print** a string (str).

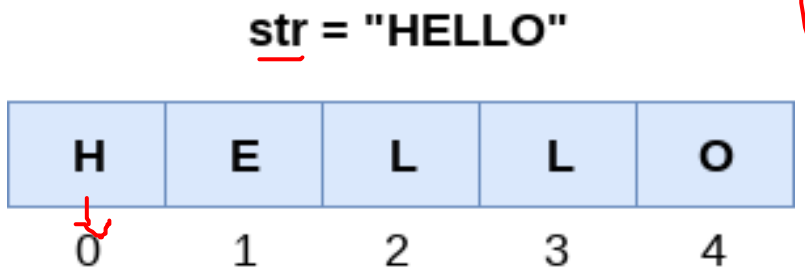
```
#Using single quotes
str1 = 'Hello Python'
print(str1)
#Using double quotes
str2 = "Hello Python"
print(str2)

#Using triple quotes
str3 = """Triple quotes are generally used for

    represent the multiline or
    docstring"""
print(str3)
```

Output:

```
Hello Python
Hello Python
Triple quotes are generally used for
    represent the multiline or
docstring
```



Strings indexing and splitting Like other languages, the indexing of the Python strings starts from 0. For example, The string "HELLO" is indexed



str[0] = 'H'

str[1] = 'E'

str[2] = 'L'

str[3] = 'L'

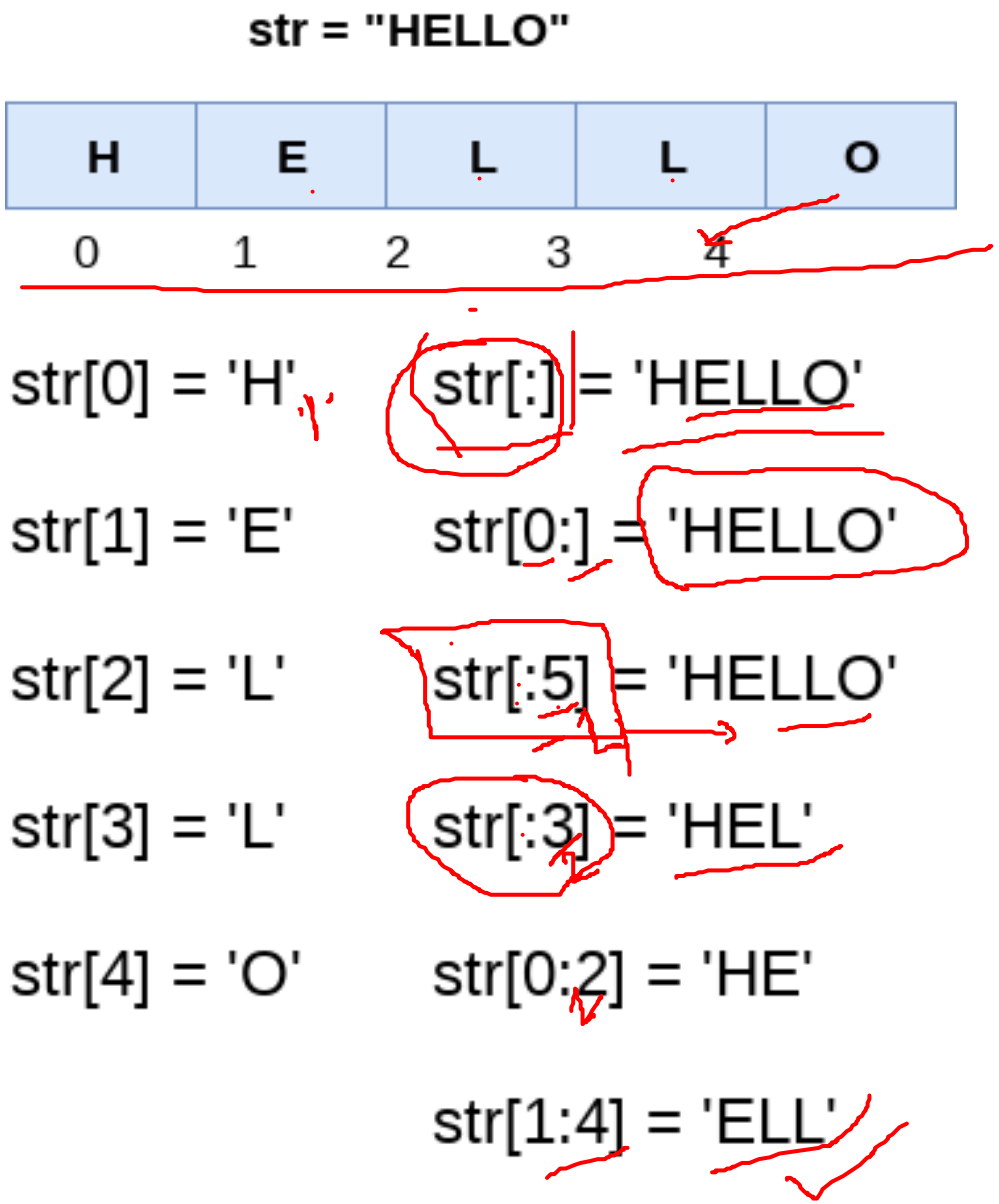
str[4] = 'O'

```
str = "HELLO"
print(str[0])
print(str[1])
print(str[2])
print(str[3])
print(str[4])
# It returns the IndexError because 6th index doesn't exist
print(str[6])
```

Output:

```
H
E
L
L
O
IndexError: string index out of range
```

- The slice operator [] is used to access the individual characters of the string. However, we can use the : (colon) operator in Python to access the substring from the given string

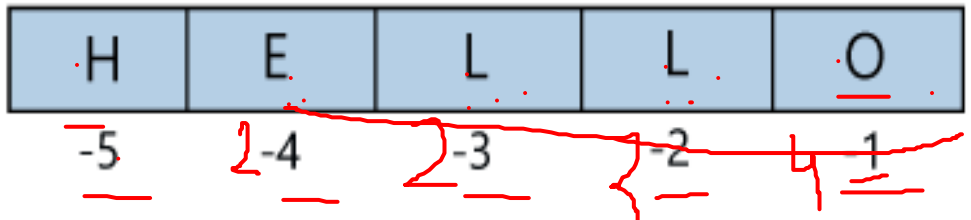


```
1.# Given String
2.str = "CE12-PYTHON"
3.# Start Oth index to end
4.print(str[0:])
5.# Starts 1th index to 4th index
6.print(str[1:5])
7.# Starts 2nd index to 3rd index
8.print(str[2:4])
9.# Starts 0th to 2nd index
10.print(str[:3])
11.#Starts 4th to 6th index
12.print(str[4:7])
```

- Output
- CE12-PYTHON
- E12-
- 12
- CE1
- -PY

We can do the negative slicing in the string; it starts from the rightmost character, which is indicated as -1.

str = “HELLO”



str[-1] = ‘O’

str[-2] = ‘L’

str[-3] = ‘L’

str[-4] = ‘E’

str[-5] = ‘H’

str[-3:-1] = ‘LL’

str[-4:-1] = ‘ELL’

str[-5:-3] = ‘HE’

str[-4:] = ‘ELLO’

str[::-1] = ‘OLLEH’

```
str = 'PYTHONFORALL'
print(str[-1])
print(str[-3])
print(str[-2:])
print(str[-4:-1])
print(str[-7:-2])
# Reversing the given string
print(str[::-1])
print(str[-12])
```

Reassigning Strings

```
str = "HELLO"
str[0] = "h"
print(str)
```

Output:

```
Traceback (most recent call last):
  File "12.py", line 2, in <module>
    str[0] = "h";
TypeError: 'str' object does not support
item assignment
```

```
str = "HELLO"
print(str)
str = "hello"
print(str)
```

Output:

```
HELLO
hello
```

Deleting the String

```
str = ”PYTHON"
del str[1]
error
```

we are deleting entire string.

```
str1 = "JAVATPOINT"
del str1
print(str1)
```

Output:

```
NameError: name 'str1' is not defined
```

String Operators

Operator	Description
+	It is known as concatenation operator used to join the strings given either side of the operator.
*	It is known as repetition operator. It concatenates the multiple copies of the same string.
[]	It is known as slice operator. It is used to access the sub-strings of a particular string.
[:]	It is known as range slice operator. It is used to access the characters from the specified range.
in	It is known as membership operator. It returns if a particular sub-string is present in the specified string.
not in	It is also a membership operator and does the exact reverse of in. It returns true if a particular substring is not present in the specified string.
%	It is used to perform string formatting. It makes use of the format specifiers used in C programming like %d or %f to map their values in python. We will discuss how formatting is done in python.

```
str = "Hello"
str1 = " world"
print(str*3) # prints HelloHelloHello
print(str+str1)# prints Hello world
print(str[4]) # prints o
print(str[2:4]); # prints ll
print('w' in str) # prints false as w is not p
resent in str
print('wo' not in str1) # prints false as wo
 is present in str1.
print(r'C://python37') # prints C://python
37 as it is written
print("The string str : %s"%(str)) # prints
 The string str : Hello
```

Output:

```
HelloHelloHello
Hello world
o
ll
False
False
C://python37
The string str : Hello
```

Python String Formatting Escape Sequence

They said, "Hello what's going on?"- the given statement can be written in single quotes or double quotes but it will raise the **SyntaxError** as it contains both single and double-quotes.

```
str = "They said, "Hello what's going on?""
print(str)
```

Output:

```
SyntaxError: invalid syntax
```

The backslash(/) symbol denotes the escape sequence. The backslash can be followed by a special character and it interpreted differently. The single quotes inside the string must be escaped.

```
# using triple quotes
print('''They said, "What's there?"''')

# escaping single quotes
print('They said, "What\'s going on?"')

# escaping double quotes
print("They said, \"What's going on?\"")
```

```
They said, "What's there?"
They said, "What's going on?"
They said, "What's going on?"
```