PYTHON PROGRAMMING

UNIT 3

Python Complex data types

By

Pragya Rajvanshi

LEC-1

Today's target

- String
- some built in function of string

Gateway classes

String

- A String represents a sequence of characters.
- It is an immutable data type(you have created a string, you cannot change it.)
- Python Programming does not have a character data type, a single character is simply a string with a length of 1.

```
1 s = "pythonprogramming"
2 print(s)
3 print(type(s))
4 z='hi'
5 print(z)
6 print(type(z))
```



```
pythonprogramming
<class 'str'>
hi
<class 'str'>
```

Gateway Classes

Accessing characters in Python String

```
s = "python"
#positive index
print(s[0])
print(s[2])
#negative index
print(s[-1])
print(s[-2])
#index number is out of the range
print(s[16])
print(s[-16])
```

```
p
t
n
o
Traceback (most recent call last):
  File "/home/main.py", line 9, in <module>
    print(s[-16])
IndexError: string index out of range
```

String slicing

substring = s[start : end : step]

Retrieve All Characters

```
1 s = "Hello"
2 s2 = s[:]
3 s3 = s[::]
4 print(s2)
5 print(s3)
```

```
Hello
Hello
```

Gayclasses

```python
s = "Hellopython"
#Characters from index 7 to the end
print(s[7:])
# Characters from the start up to index 5 (exclusive)
print(s[:5])
 #Characters from index 1 to index 5 (excluding 5)
print(s[1:5])
 #Every second character
print(s[::2])
# Every third character from index 1 to 8 (exclusive)
print(s[1:8:3])
#Out-of-Bounds Slicing
print(s[3:28])
```

```
thon
Hello
ello
Hloyhn
eot
lopython
```

```python
s = "abcdefghijklmno"
# Characters from index -4 to the end
print(s[-4:])
# Characters from the start up to index -3 (excluding -3)
print(s[:-3])
# Characters from index -5 to -2 (excluding -2)
print(s[-5:-2])
# Get every 2nd elements from index -8 to -1 (excluding index -1)
print(s[-8:-1:2])
```

```
lmno
abcdefghijkl
klm
hjln
```

Gateway classe

```
1 s = "Python"
2 # Reverse the string
3 print(s[::-1])
```



```
nohtyP
```

String built-in function

1. **capitalize()** method

- **It is used to change the first letter of a string to uppercase and make all other letters lowercase.**

```
#example1
s = "hello WORLD"
res = s.capitalize()
print(res)
#example2
z = "123hello WORLD"
res1 = z.capitalize()
print(res1)
```

```
Hello world
123hello world
```

Gateway classes

2. isalnum() Method

This method checks whether all the characters in a given string are either alphabet or numeric (alphanumeric) characters

```
string = "abc123"
print(string.isalnum())
s1 = "abc 123"
print(s1.isalnum())
s2 = "abc_123"
print(s2.isalnum())
s3 = "000"
print(s3.isalnum())
s4= "aaaa"
print(s4.isalnum())
```

```
True
False
False
True
True
```

3. isalpha() Method

It is used to check whether all characters in the String are an alphabet.

```
string1 = 'Ayush'
print(string1.isalpha())
string2 = 'Ayush0212'
print(string2.isalpha())
string3 = 'Ayush Saxena'
print( string3.isalpha())
string4 = ""
print(string4.isalpha())
```

```
True
False
False
False
```

Gateway classes

4. isdecimal() Method

This method checks whether all the characters in a given string is digit.

```
s1 = "12345"
print(s1.isdecimal())
s2 = "12geeks34"
print(s2.isdecimal())
s3 = "12/34"
print(s3.isdecimal())
```

```
True
False
False
```

5. isdigit() Method

It returns "True" if all characters in the string are digits, Otherwise, It returns "False"

```
s1 = '15460'
print(s1.isdigit())
s2 = '154ayush60'
print(s2.isdigit())
```

```
True
False
```

6. index() Method

The index() method in Python is used to find the position of a specified substring within a given string.

```
7
Traceback (most recent call last):
  File "/home/main.py", line 4, in <module>
    position2=s.index("yag")
ValueError: substring not found
```

```
s = "Python programming"
position = s.index("prog")
print(position)
position2=s.index("yag")
print(position2)
```

Gateway classes

Syntax of index()

s.index(substring, start=0, end=len(s))

1. **substring:** The substring to locate within the string s.
2. **start (optional):** The starting index for the search. Defaults to 0 if not provided.
3. **end (optional):** The ending index for the search. If not provided, it defaults to the length of the string.

```
s = "Python"
position = s.index("on", 1, 6)
print(position)
```

```
4
```

7. islower() method

This method checks if all characters in the string are lowercase.

```
print("python".islower())
print("pYthon".islower())
```

```
True
False
```

8. isnumeric() Method

It is used to determine whether the string consists of numeric characters or not

```
string1 = '123ayu456'
print(string1.isnumeric())
string2 = '123456'
print(string2.isnumeric())
```

```
False
True
```

Gateway classes

9. isidentifier() Method

- This method is used to check whether a string is a valid identifier or not

Note:

A string is considered as a valid identifier if:

- It only consists of alphanumeric characters and underscore (_)
- Doesn't start with a space or a number

```
# String with spaces
string = "hi for G"
print(string.isidentifier())
# A Perfect identifier
string = "hellopython"
print(string.isidentifier())
# Empty string
string = ""
print(string.isidentifier())
# Alphanumerical string
string = "hi0for0hi"
print(string.isidentifier())
# Beginning with an integer
string = "54k0for0ks"
print(string.isidentifier())
```

```
False
True
False
True
False
```

Gateway Classes

10. upper() Method

- It creates a new string with all lowercase letters changed to uppercase.
- This method does not change the original string instead it simply returns a new one.

```
s = "hello, world!"
res = s.upper()
print(res)
```

```
HELLO, WORLD!
```

11. isupper() method

This method returns whether all characters in a string are uppercase or not.

```
a= "HELLO"
b= "hello"
c="hEllo"
print(a.isupper())
print(b.isupper())
print(c.isupper())
```

```
True
False
False
```

12. lower() Method

It converts all uppercase letters in a string to their lowercase.

```
1 s = "HELLO, WORLD!"
2 res = s.lower()
3 print(res)
```

```
hello, world!
```

Gateway classes



Gateway Classes