Fundamentals of Python

PYTHON STRINGS

Shruti Bharat

@shrutibharat0105

Table of Contents:

Creating a string

In Python, a string is a sequence of characters enclosed within single, double, or triple quotes.

```
# Single quote
string1 = 'Hello, World!'

# Double quote
string2 = "Hello, World!"

# Triple quotes for multi-line strings
string3 = '''Hello,
World!'''

print(string1) # Hello, World!
print(string2) # Hello, World!
print(string3) # Hello, World!
```

Accessing substrings from a string

Substrings can be accessed using indexing and slicing. Indexing starts at 0 for the first character.
There are 2 types of indexing:

1. Positive indexing
2. Negative indexing

```
string = "Hello, World!"

# Indexing
# Positive indexing
print(string[1])  # e

# Negative indexing
print(string[-1])  # !

# Slicing a substring
# Positive slicing
print(string[0:5])  # Hello

# Negative slicing(1st element should be greater than 2nd)
print(string[6:0:-1])  # W olle

# Reversing a string
print(string[::-1]) # !dlroW olleH
```

Deleting a string

Strings in Python are immutable, so you cannot change them directly.

However, you can create new strings based on operations. Deleting can be done using del for variables.

Immutability refers to an object's inability to be changed after it has been created.

```
string = "Hello, World!"

# Deleting a string variable
del string

# This will give you error
# As strings are immutable you can't partially delete
# Only whole string can be delete not partial
del string[-1:-5:2]
```

Operations on a string

Various operations like arithmetic, loops on strings, and membership operators can be performed on strings.

```
string1 = "Hello"
string2 = "World"

# Arithmatic
# Addition
concat = string1 + string2
print(concat)  # HelloWorld

# Multiplication
repeat = string1 * 3
print(repeat)  # HelloHelloHello

# Loops on string
for i in range 'string1':
  print('Hi') # HiHiHiHiHiHiHi

# Membership
# in, not in operator
print("Hello" in concat)  # True
print("Hello" not in concat) # False
```

String Functions

Python provides many built-in functions to manipulate strings

```
string = "hello, world!"

# 1. Length of string
print(len(string))  # Output: 13

# 2. Maxium of string(It is based on unicode value of char)
print(max(string))  # Output: w

# 3. Minmum of string(It is based on unicode value of char)
print(min(string))  # Output: ' '

# 4. Sorted(temporarily sorts the string)
# By default it is in ascending order
print(sorted(string)) # Output: [' ','d','e','h','l','l',
                                # 'o','o','r','w']

# Descending order
print(sorted(string, reverse = True))

# 5. Convert to Lowercase
print(string.lower())  # Output: "hello, world!"

# 6. Convert to Uppercase
print(string.upper())  # Output: "HELLO, WORLD!"
```

String Functions

```python
# 7. Capitalize String
print(string.capitalize())  # Output: "Hello, world!"

# 8. Title Case String
print(string.title())  # Output: "Hello, World!"

# 9. Startswith
print(string.startswith("Hello"))  # Output: True

# 10. Endswith
print(string.endswith("!"))  # Output: True

# 11. Find Substring
print(string.find("World"))  # Output: 7

# 12. Index of Substring
print(string.index("World"))  # Output: 7

# 13. Count Substring
print(string.count("o"))  # Output: 2

# 14. Strip Whitespace
whitespace_string = "   Hello, World!   "
print(whitespace_string.strip())  # Output: "Hello, World!"
```

String Functions

```python
# 15. Replace Substring
print(string.replace("World", "Python"))
# Output: "Hello, Python!"

# 16. Split String
print(string.split(", "))  # Output: ["Hello", "World!"]

# 17. Join List of Strings
list_of_strings = ["Hello", "World"]
print(" ".join(list_of_strings))  # Output: "Hello World"

# 18. Check Alphabetic Characters
alpha_string = "Hello"
print(alpha_string.isalpha())  # Output: True

# 19. Check Digits
digit_string = "12345"
print(digit_string.isdigit())  # Output: True

# 20. Check Alphanumeric Characters
alnum_string = "Hello123"
print(alnum_string.isalnum())  # Output: True
```

Coding problems

Python String Exercises

- Python program to check whether the string is Symmetrical or Palindrome
- Reverse words in a given String in Python
- Ways to remove i’th character from string in Python
- Find the length of a string in Python (4 ways)
- Python program to print even length words in a string

Bonus Tips

String formatting in Python is the process of embedding variables and expressions within string literals to create dynamic and readable text output.

```
name = "John"
age = 30

# Common Specifiers
pi_value = 3.14159
new_str = "Pi is approximately %.2f." % pi_value
print(new_str)  # Output: Pi is approximately 3.14.

# str.format() Method
new_Str1 = "My name is {} and I am {} years old.".format(name,age)
print(new_str1)  # Output: My name is John and I am 30 years old.

# Positional and Keyword Arguments
new_str2 = "My name is {0} and I am {1} years old.".format(name,age)
print(new_str2)  # Output: My name is John and I am 30 years old.

new_str3 = "My name is {name} and I am {age} years old.".format(name=name, age=age)
print(new_str3)  # Output: My name is John and I am 30 years old.

# F-Strings (Formatted String Literals)
new_str4 = f"My name is {name} and I am {age} years old."
print(new_Str4)  # Output: My name is John and I am 30 years old.
```

Bonus Tips:

A **raw string** in Python is a string prefixed with r or R, which treats backslashes \ as literal characters, ignoring their escape functionality.

```
# Without raw string
regular_string = "C:\new\text.txt"
print(regular_string)  # Output: C:
# ew  ext.txt (backslashes interpreted as escape characters)

# With raw string
raw_string = r"C:\new\text.txt"
print(raw_string)
# Output: C:\new\text.txt (backslashes treated as
# literal characters)
```

🚀 Ready to Level Up?

CONNECT FOR MORE INSIGHTS AND FUTURE UPDATES!

Shruti Bharat

@shrutibharat0105