Strings

Seq. of Chars

In Python: *Unicode Chars*

Ops:

- Create
- Access
- Add Chars
- Edit
- Delete
- Ops on Strings
- String Functions

1. Create ¶

In [1]:
```
a = 'Hello'
print(a)
```

```
Hello
```

In [2]:
```
c = "Hello"
print(c)
```

```
Hello
```

In [4]:
```
'It"s raining outside'
```

Out[4]:
```
'It"s raining outside'
```

In [4]:
```
"It's raining outside"
```

Out[4]: "It's raining outside"

In [5]:
```
c = '''Hello,
How are you boi''' # multi-line string
print(c)
```

```
Hello,
How are you boi
```

In [6]:
```
c = """Hello"""
print(c)
```

```
Hello
```

In [7]:
```
c = str('Hello')
c
```

Out[7]: 'Hello'

2. Access

Accessing Substrings from a String

In [13]:
```
# Concept of Indexing
c = "hello"
print(c)
```

```
hello
```

In [14]:
```
c[0]
```

Out[14]: 'h'

Indexing Types:

- Positive
- Negative

In [20]:
```
print(c[-5])
```

```
h
```

In [60]:
```
# Slicing
c = "Hello World"
print(c)
```

```
Hello World
```

In [23]:
```
print(c[0:11:2]) # start index and end index -1
```

```
HloWrd
```

In [22]:
```
print(c[:6])
```

```
Hello
```

In [23]:
```
print(c[6:])
```

```
Hello World
```

In [24]:
```
print(c[::-1])
```

```
dlroW olleH
```

In [25]:
```
print(c[0:11:2])
```

```
HloWrd
```

In [38]:
```
print(c[0:10:3])
```

```
HlWl
```

In [3]:
```
print(c[::-1])
```

```
dlroW olleH
```

3. Add Chars

In [26]:
```
string_1 = "Hello"
string_2 = 'World'
```

In [28]:
```
new_string = string_1 + " " + string_2
new_string
```

Out[28]:
```
'Hello World'
```

In [29]:
```
another_string = string_1 + ", how are you?"
another_string
```

Out[29]:
```
'Hello, how are you?'
```

4. Edit & 5. Delete

In [30]:
```
c = "hello"
print(c)
```

```
hello
```

In [32]:
```
c[0] = "X"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[32], line 1
----> 1 c[0] = "X"

TypeError: 'str' object does not support item assignment
```

Strings are a Immutable Data Types

In [33]:
```
c = "World"
print(c)
```

```
World
```

In [34]:
```
c[4] = "X"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[34], line 1
----> 1 c[4] = "X"

TypeError: 'str' object does not support item assignment
```

In [35]:
```
# Deletion
c
```

Out[35]:
```
'World'
```

In [36]:
```
del c
```

In [37]:
```
print(c)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[37], line 1
----> 1 print(c)

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

In [43]:
```
c = "hello"
```

In [44]:
```
print(c)
```

```
hello
```

In [45]:
```
del c[1]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[45], line 1
----> 1 del c[1]

TypeError: 'str' object doesn't support item deletion
```

In [46]:
```
del c[:3:2]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[46], line 1
----> 1 del c[:3:2]

TypeError: 'str' object does not support item deletion
```

6. Ops on Strings

- **Arithmetic**: Concatenation (+), Repetition (*)
- **Relational**: Equality (==), Inequality (!=), Comparison (< , > , <= , >=)
- **Logical**: `and` , `or` , `not`
- **Loops**: `for` loop, `while` loop
- **Membership**: `in` , `not in`

In [47]:
```
# Arithmetic
"Hello" + " - " + "World"
```

Out[47]: 'Hello - World'

In [48]:
```
print("*" * 100)
```

```
****************************************************************************************************
```

In [49]:
```
print("Hello"* 20)
```

```
HelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHelloHello
```

In [50]:
```
# Relational
"Hello" == "World"
```

Out[50]: False

In [51]:
```
"Hello" != "World"
```

Out[51]: True

In [54]:
```
"Mumbai" > "Pune" # Lexiographically
```

Out[54]: False

In [55]:
```
"Goa" > "Kolkata"
```

Out[55]: False

In [56]:
```
"kol" < "Kol"
```

Out[56]: False

In [52]:
```
# Logical
"Hello" and "World"
```

Out[52]: 'World'

In [43]:
```
# ""        ---> False
# "saurabh" ---> True
```

In [53]:
```
"Hello" and ""
```

Out[53]: ''

In [54]:
```
"" or "World"
```

Out[54]: 'World'

In [55]:
```
"Hello" or "World"
```

Out[55]: 'Hello'

In [56]:
```
"Hello" and "World"
```

Out[56]: 'World'

In [62]:
```
not "Hello"
```

Out[62]: False

In [63]:
```
print(not "Hello")
```

```
False
```

In [64]:
```
not ""
```

Out[64]:
```
True
```

In [65]:
```
# Loops on Strings
c = "Hello World"
for i in c:
    print(i)
```

```
H
e
l
l
o

W
o
r
l
d
```

In [66]:
```
for i in c[2:7]:
    print(i)
```

```
l
l
o

W
```

In [67]:
```
for i in c[2:7:2]:
    print(i)
```

```
l
o
W
```

In [59]:
```
for i in c[::-1]:
    print(i)
```

```
d
l
r
o
W

o
l
l
e
H
```

In [61]:
```
# Membership
c
```

Out[61]: 'Hello World'

In [63]:
```
'H' in c
```

Out[63]: True

In [64]:
```
'h' in c
```

Out[64]: False

In [65]:
```
'World' not in c
```

Out[65]: False

7. String Functions

1. Common Functions

- len()
- max()
- min()
- sorted()

In [66]:
```
c = 'Mumbai'
len(c)
```

Out[66]: 6

In [67]:
```
max(c)
```

Out[67]: 'u'

In [68]:
```
min(c)
```

Out[68]: 'M'

In [69]:
```
sorted(c)
```

Out[69]: ['M', 'a', 'b', 'i', 'm', 'u']

In [72]:
```
sorted(c, reverse=True)
```

Out[72]: ['u', 'm', 'i', 'b', 'a', 'M']

2. Capitalize/Title/Upper/Lower/Swapcase

In [8]:
```
c = "mumbai"
c.capitalize()
```

Out[8]: 'Mumbai'

In [9]:
```
c
```

Out[9]: 'mumbai'

In [10]:
```
'the weather is pretty cold today'.capitalize()
```

Out[10]: 'The weather is pretty cold today'

In [11]:
```
'the weather is pretty cold today'.title()
```

Out[11]: 'The Weather Is Pretty Cold Today'

In [12]:
```
c.upper()
```

Out[12]: 'MUMBAI'

In [13]:
```
c.lower()
```

Out[13]: 'mumbai'

In [18]:
```
"mumBAI".swapcase()
```

Out[18]: 'MUMbai'

3. Count

In [20]:
```
"it is raining".count("i")
```

Out[20]: 4

In [21]:
```
"it is raining".count("ing")
```

Out[21]: 1

In [28]:
```
"it is raining".count("X")
```

Out[28]: 0

4. Find/Index

In [60]:
```
"My name is Shyam".find("i")
```

Out[60]: 8

In [23]:
```
"it is raining".find("i")
```

Out[23]: 0

In [24]:
```
"it is raining".find("raining")
```

Out[24]: 6

In [25]:
```
"it is raining".find("x")
```

Out[25]: -1

In [26]:
```
"it is raining".index("s")
```

Out[26]:
```
4
```

In [27]:
```
"it is raining".index("x")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[27], line 1
----> 1 "it is raining".index("x")

ValueError: substring not found
```

5. endswith/startswith

In [29]:
```
"Today we are starting with python and the topic goin to be is string".endswith("ing")
```

Out[29]:
```
True
```

In [30]:
```
"Today we are starting with python and the topic goin to be is string".endswith("we")
```

Out[30]:
```
False
```

In [33]:
```
"it is raining".startswith("it")
```

Out[33]:
```
True
```

6. format

In [35]:
```
"Today is {} and the weather is too {}".format(9,"warm")
```

Out[35]:
```
'Today is 9 and the weather is too warm'
```

In [38]:
```
"Today is {1} and the weather is too {0}".format(9,"warm")
```

Out[38]:
```
'Today is warm and the weather is too 9'
```

In [42]:
```
"Hello my name is {name} and I am {age} years old".format(name = "Harshad", age = 7)
```

Out[42]:
```
'Hello my name is Harshad and I am 7 years old'
```

In [45]:
```
"Hello my name is {age} and I am {name} years old".format(name = "Bushan",age = )
```

Out[45]:
```
'Hello my name is Bushan and I am Bushan years old'
```

In [48]:
```
"Hello my name is {name} and I am {age} year old ".format(name = "Saurabh", age = 22, weight = 90)
```

Out[48]:
```
'Hello my name is Saurabh and I am 22 year old '
```

In [47]:
```
name = "Umar"
name_1 = "Sahil"
print(f"Hello!...{name}, how are you..... Am good {name_1}, thanks for asking.")
```

```
Hello!...Umar, how are you..... Am good Sahil, thanks for asking.
```

7. isalnum/isalpha/isdecimal/isdigit/isidentifier

In [59]:
```
"FLAT20".isalnum() # Alphanumeric
```

Out[59]:
```
True
```

In [60]:
```
"FLAT20&".isalnum()
```

Out[60]:
```
False
```

In [41]:
```
"FLAT".isalpha() # Alphabetic
```

Out[41]:
```
True
```

In [42]:
```
"FLAT20".isalpha()
```

Out[42]: False

In [43]:
```
"20".isdigit()
```

Out[43]: True

In [44]:
```
"20A".isdigit()
```

Out[44]: False

In [51]:
```
"Hello World".isidentifier()
```

Out[51]: False

In [52]:
```
"Hello_World".isidentifier()
```

Out[52]: True

8. Split

In [53]:
```
"who is the pm of india".split()
```

Out[53]: ['who', 'is', 'the', 'pm', 'of', 'india']

In [54]:
```
"who is the pm of india".split("pm")
```

Out[54]: ['who is the ', ' of india']

In [29]:
```
"who is the pm of india".split("i")
```

Out[29]: ['who ', 's the pm of ', 'nd', 'a']

In [30]:
```
"who is the pm of india".split("y")
```

Out[30]:
```
['who is the pm of india']
```

9. Join

In [3]:
```
" ".join(['who', 'is', 'the', 'pm', 'of', 'india'])
```

Out[3]:
```
'who is the pm of india'
```

In [58]:
```
"-".join(['who', 'is', 'the', 'pm', 'of', 'india'])
```

Out[58]:
```
'who-is-the-pm-of-india'
```

10. Replace

In [14]:
```
"Hi my name is Sandy".replace("Sandy", "sid")
```

Out[14]:
```
'Hi my name is sid'
```

11. Strip

In [59]:
```
name = "          Rohan          "
name
```

Out[59]:
```
'          Rohan          '
```

In [62]:
```
"Hi Good morning" + name
```

Out[62]:
```
'Hi Good morning          Rohan          '
```

In [65]:
```
"Hi Good morning " + name.strip()
```

Out[65]:
```
'Hi Good morning Rohan'
```

Example Programs

In [2]:
```
# 1. Length of String without len()
s = input('enter the string: ')
counter = 0
for i in s:
    counter += 1
print('length of string is', counter)
```

```
enter the string: saurabh
length of string is 7
```

In [16]:
```
# 2. Extract username from email
# Eg: saurabhsinghdhami@gmail.com ---> saurabhsinghdhami
s = input('enter the email: ')
pos = s.index('@')
print(s[0:pos])
```

```
enter the email: saurabhsinghdhami@gmail.com
saurabhsinghdhami
```

In [25]:
```
# 3. Count character frequency in a string
s = input('enter the email: ')
term = input('what would like to search for: ')
counter = 0
for i in s:
  if i == term:
    counter += 1
print('frequency', counter)
```

```
enter the email: alibaig@vorizon.com
what would like to search for: a
frequency 2
```

In [1]:
```python
# 4. Remove Character from String
s = input('enter the string: ')
term = input('what would like to remove: ')
result = ''
for i in s:
  if i != term:
    result += i
print(result)
```

```
enter the string: my name is ali
what would like to remove: a
my nme is li
```

In [18]:
```python
# 5. Check if a string is a palindrome
s = input('enter the string: ')
flag = True
for i in range(0,len(s)//2):
  if s[i] != s[len(s) - i -1]:
    flag = False
    print('Not a Palindrome')
    break
if flag:
  print('Palindrome')
```

```
enter the string: maam
Palindrome
```

In [29]:
```
# 6. Word Count Without split()
s = input('enter the string: ')
L = []
temp = ''
for i in s:
  if i != ' ':
    temp += i
  else:
    L.append(temp)
    temp = ''
L.append(temp)
print(L)
```

```
enter the string: My name is Ramo D seuza
['My', 'name', 'is', 'Ramo', 'D', 'seuza']
```

In [7]:
```
# 7. Convert String to Title Case (No title())
s = input('enter the string: ')
L = []
for i in s.split():
    L.append(i[0].upper() + i[1:].lower())
print(" ".join(L))
```

```
enter the string: hello! the day is pretty awesome
Hello! The Day Is Pretty Awesome
```

In [31]:
```
# 8. Integer to String Conversion
number = int(input('enter the number: '))
digits = '0123456789'
result = ''
while number != 0:
  result = digits[number % 10] + result
  number //= 10
print(result)
print(type(result))
```

```
enter the number: 12345678910
12345678910
<class 'str'>
```

In [ ]: