APSSDC

Andhra Pradesh State Skill Development Corporation

(https://apssdc.in)

Day04 Online Training on Python Programming

Day04 Objectives

- Operators Contd.
  - Assignment Operators
  - Bitwise Operators
  - Membership Operators
  - Identity Operators
- Strings
  - What and How to declare a string
  - Accessing the Elements from the string
  - String Methods

Assignment Operators

In [1]:
```
a = 5
b = 10
```

In [4]:
```
a = a + 10

print(a)
```

```
25
```

In [5]:
```
a += 10

print(a)
```

```
35
```

In [7]:
```
a -= 10

print(a)
```

```
15
```

In [10]:

```
a *= 10
print(a)
a /= 5
print(a)
a //= 2
print(a)
a **= 5
print(a)
a %= 3
print(a)
```

```
150
30.0
15.0
759375.0
0.0
```

Bitwise Operators

- Bitwise AND `&` - inp1 * inp2
- Bitwise OR `|` - inp1 + inp2
- Bitwise NOT `~` - inp1'
- Bitwise XOR `^` - inp1' * inp2 + inp1 * inp2'
- Right Shift `>>`
- Left Shift `<<`

In [2]:

```
a = 10 # 1010
b = 7 # 0111
```

```
1010
0111
0010 --> 2
```

In [13]:

```
a & b
```

Out[13]: 2

1010 0111

1111 --> A | B

10001 --> Binary Addition

In [14]:
```
a | b
```

Out[14]: 15

In [3]:
```
~a # 1010 -(a+1) - 1011

```

Out[3]: -11

In [4]:
```
c = -15

~c
```

Out[4]: 14

```
1010
0111

1101
|
0 * 0 + 1 * 1 --> 1

0 0 -> 0 A'B + AB'
1 0 -> 1
0 1 -> 1
1 1 -> 0
```

In [5]:
```
a ^ b # 1010 ^ 0111
```

Out[5]: 13

Right Shift Operator

In [6]:
```
a = 15 # 1111
```

```
a >>2 => 1111 >> 2
      => 0011
      => 3
```

In [7]:
```
a >> 2
```

Out[7]: 3

Left Shift Operator

In [10]:
```
a << 2 # 111100
```

Out[10]: 60

Special Operators in python

Membership Operator

- in
- not in

In [11]:
```
e1 = 'apssdc@gmail.com'
e2 = 'apssdc@srkit.edu.in'
```

In [12]:
```
'srkit' in e1
```

Out[12]: False

In [13]:
```
'srkit' not in e1
```

Out[13]: True

identity operator

- is
- is not

In [17]:
```
a = 5
b = 5

c = ['apssdc', '123', 'SRKIT']
d = ['apssdc', '123', 'SRKIT']
e = d
```

In [15]:
```
id(a), id(b)
```

Out[15]: (140721362122656, 140721362122656)

In [18]:
```
id(c), id(d), id(e)
```

Out[18]: (1899834063168, 1899834274560, 1899834274560)

In [19]:
```
a is b
```

Out[19]: True

In [20]:
```
c is d
```

Out[20]: False

In [22]:
```
d is e
```

Out[22]: True

In [23]:
```
s1 = 'APSSDC'
s2 = 'APSSDC'

id(s1), id(s2)
```

Out[23]: (1899835287088, 1899835287088)

Strings

- Sequence of characters
- Unicode encoding strings
- Strings are immutable --> Update or Delete the characters inside the string

In [25]:
```
s1 = 'APSSSDC'
type(s1)
```

Out[25]: str

In [26]:
```
s2 = "APSSDC"
type(s2)
```

Out[26]: str

In [28]:
```
s3 = """APSSDC
python
training
program
for
SRKIT"""
```

In [29]:
```
s4 = '''APSSDC
python
training
program
for
SRKIT'''
```

In [30]:
```
sd = ''
sdd = str()
```

In [31]:
```
1 type(sd), type(sdd)
```

Out[31]: (str, str)

Accessing the Elements from the String / Indexing

In [33]:
```
1 print(s3)
```

```
APSSDC
python
training
program
for
SRKIT
```

In [34]:
```
1 s3
```

Out[34]: 'APSSDC\npython\ntraining\nprogram\nfor\nSRKIT'

In [35]:
```
1 s3[0]
```

Out[35]: 'A'

In [36]:
```
1 s3[-1]
```

Out[36]: 'T'

In [37]:
```
1 s3[5:10]
```

Out[37]: 'C\npyt'

In [38]:
```
1 s3[-5:-1] # From Last 5th char to -1-1 --> -2 characted
```

Out[38]: 'SRKI'

Syntax for indexing

```
string_variable[start : end : step] # ending index is excluded
```

In [40]:
```
1 s3[: :]
```

Out[40]: 'APSSDC\npython\ntraining\nprogram\nfor\nSRKIT'

In [41]:
```
1 s3[: : 2]
```

Out[41]: 'ASD\nyhntann\nrga\no\nRI'

In [42]:
```
1 s3[: : -2]
```

Out[42]: 'TKSrfmropgiir\notpCSP'

In [43]:
```
1 s3[: : 5]
```

Out[43]: 'AChrno\nS'

In [44]:
```
1 s3[: : 1]
```

Out[44]: 'APSSDC\npython\ntraining\nprogram\nfor\nSRKIT'

In [45]:
```
1 s3[: : -1]
```

Out[45]: 'TIKRS\nrof\nmargorp\ngniniart\nnohtyp\nCDSSPA'

Arthematic Operators for strings

In [52]:
```
1 s1
```

Out[52]: 'APSSSDC Python Python'

In [53]:
```
1 s1 + ' Python'
```

Out[53]: 'APSSSDC Python Python Python'

In [54]:
```
1 s1
```

Out[54]: 'APSSSDC Python Python'

In [55]:
```
1 s1 += ' Python'
2
3 s1
```

Out[55]: 'APSSSDC Python Python Python'

In [56]:
```
1 s2
```

Out[56]: 'APSSDC'

In [58]:
```
1 s2 += ' '
2 s2 * 5
```

Out[58]: 'APSSDC APSSDC APSSDC APSSDC APSSDC '

In [59]:
```
1 s2
```

Out[59]: 'APSSDC '

In [60]:
```
1 s1 + s2
```

Out[60]: 'APSSSDC Python Python PythonAPSSDC '

In [61]:
```
1 s1 + 10
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-61-2c33e8bda66d> in <module>
----> 1 s1 + 10

TypeError: can only concatenate str (not "int") to str
```

In [63]:
```
1 s1 + " " + str(10)
```

Out[63]: 'APSSSDC Python Python Python 10'

String Methods

In [64]:
```
1 string = 'Day04 session of Python Programming'
```

In [65]:
```
1 dir(string)
```

Out[65]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mod__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmod__',
 '__rmul__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'capitalize',
 'casefold',
 'center',
 'count',
 'encode',
 'endswith',
 'expandtabs',
 'find',
 'format',
 'format_map',
 'index',
 'isalnum',
 'isalpha',
 'isascii',
 'isdecimal',
 'isdigit',
 'isidentifier',
 'islower',
 'isnumeric',
 'isprintable',
 'isspace',
```

```
 'istitle',
 'isupper',
 'join',
 'ljust',
 'lower',
 'lstrip',
 'maketrans',
 'partition',
 'replace',
 'rfind',
 'rindex',
 'rjust',
 'rpartition',
 'rsplit',
 'rstrip',
 'split',
 'splitlines',
 'startswith',
 'strip',
 'swapcase',
 'title',
 'translate',
 'upper',
 'zfill']
```

In [66]:
```
1 help(str.count)
```

```
Help on method_descriptor:

count(...)
    S.count(sub[, start[, end]]) -> int

    Return the number of non-overlapping occurrences of substring sub in
    string S[start:end].  Optional arguments start and end are
    interpreted as in slice notation.
```

In [67]:
```
1 string.upper()
```

Out[67]: 'DAY04 SESSION OF PYTHON PROGRAMMING'

In [68]:
```
1 string.lower()
```

Out[68]: 'day04 session of python programming'

In [69]:
```
1 string.casefold()
```

Out[69]: 'day04 session of python programming'

In [70]:
```
1 string.swapcase()
```

Out[70]: 'dAY04 SESSION OF pYTHON pROGRAMMING'

In [71]:
```
1 string.isupper()
```

Out[71]: False

In [72]:
```
1 string
```

Out[72]: 'Day04 session of Python Programming'

In [73]:
```
1 string[0].isupper()
```

Out[73]: True

In [75]:
```
1 string.islower()
```

Out[75]: False

In [76]:
```
1 string[-5:].islower()
```

Out[76]: True

In [77]:
```
1 string.isalpha()
```

Out[77]: False

In [78]:
```
1 string.isalnum() # checking all the chars in string are numericals and a
```

Out[78]: False

In [80]:
```
1 ord('0'), ord('9')
```

Out[80]: (48, 57)

In [82]:
```
1 mobile = '0123456789ABC'
2
3 mobile.isalnum()
```

Out[82]: True

In [83]:
```
1 mobile = '1234567890'
2
3 mobile.isdigit()
```

Out[83]: True

In [84]:
```
1 string.isdigit()
```

Out[84]: False

In [85]:
```
space = '    '

space.isspace()
```

Out[85]: True

In [88]:
```
tit = string.title()
```

In [87]:
```
string
```

Out[87]: 'Day04 session of Python Programming'

In [89]:
```
tit.istitle()
```

Out[89]: True

In [90]:
```
string.istitle()
```

Out[90]: False

In [91]:
```
string.capitalize()
```

Out[91]: 'Day04 session of python programming'

In [92]:
```
s3
```

Out[92]: 'APSSDC\npython\ntraining\nprogram\nfor\nSRKIT'

In [93]:
```
s3.capitalize()
```

Out[93]: 'Apssdc\npython\ntraining\nprogram\nfor\nsrkit'

In [94]:
```
string
```

Out[94]: 'Day04 session of Python Programming'

In [95]:
```
string.count('s')
```

Out[95]: 3

In [96]:
```
string.count('on')
```

Out[96]: 2

In [97]:
```
string.split()
```

Out[97]: ['Day04', 'session', 'of', 'Python', 'Programming']

In [98]:
```
1 e1
```

Out[98]: 'apssdc@gmail.com'

In [101]:
```
1 e1.split('@')
```

Out[101]: ['apssdc', 'gmail.com']

In [102]:
```
1 e1.split('.')
```

Out[102]: ['apssdc@gmail', 'com']

In [105]:
```
1 e3 = 'apssdc.ml.python@gmail.com'
2
3 e3.split('.')
```

Out[105]: ['apssdc', 'ml', 'python@gmail', 'com']

In [106]:
```
1 a = '   abc   '
2 y = '(APSSDC)'
```

In [107]:
```
1 a.strip()
```

Out[107]: 'abc'

In [112]:
```
1 roll = '15aps2020 - apssdc'
2
3 roll.split('-')[0].strip().upper()
```

Out[112]: '15APS2020'

In [113]:
```
1 a.rstrip()
```

Out[113]: '   abc'

In [114]:
```
1 a.lstrip()
```

Out[114]: 'abc   '

In [115]:
```
1 y.strip('()')
```

Out[115]: 'APSSDC'

In [116]:
```
1 y.rstrip('()')
```

Out[116]: '(APSSDC'

In [117]:

```
1 y.lstrip('()')
```

Out[117]: 'APSSDC)'