string

- collection of character
- string is immutable

In [5]:
```
s="python programming"
s
```

Out[5]: 'python programming'

In [6]:
```
s
```

Out[6]: 'python programming'

string slicing

In [7]:
```
s[5]
```

Out[7]: 'n'

In [8]:
```
 s[1:6]
```

Out[8]: 'ython'

In [9]:
```
s[-1]
```

Out[9]: 'g'

In [10]:
```
s[-3]
```

Out[10]: 'i'

In [11]:
```
s[len(s)-1]
```

Out[11]: 'g'

In [12]:
```
len(s)
```

Out[12]: 18

In [13]:
```
s[0:18]
```

Out[13]: 'python programming'

In [17]:
```
s[-18]
```

Out[17]: 'p'

In [18]:
```
s[len(s)//2]
```

Out[18]: 'o'

In [20]:
```
s="programming"
print(s)
print(len(s))
print(s[5])
```

```
programming
11
a
```

In [22]:
```
s1="python"
print(len(s1))
print(s1[len(s1)//2])
print(s1[len(s1)//2-1:len(s1)//2+1])
```

```
6
h
th
```

In [23]:
```
s1[::-1]
```

Out[23]: 'nohtyp'

In [25]:
```
s2="python programming"
print(len(s2))
print(s2[len(s2)//2])
print(s1[len(s2)//2-1:len(s2)//2+1])
```

```
18
o
```

In [28]:
```
print(s2)
s2[0:5:1]
```

```
python programming
```

Out[28]: 'pytho'

In [29]:
```
s2[0::3]
```

Out[29]: 'ph oai'

In [30]:
```
s2[0:5:2]
```

Out[30]: 'pto'

In [31]:
```
s3="abcdefgh"
```

In [32]:
```
s3
```

Out[32]: 'abcdefgh'

In [33]:
```
s3[0:5:2]
```

Out[33]: 'ace'

In [34]:
```
dir(str)
```

Out[34]:
```
['__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 [38]:
```
s="hello"
s.__add__("hello1")
```

Out[38]:
```
'hellohello1'
```

In [40]:
```
s.capitalize()
```

Out[40]:
```
'Hello'
```

In [41]:
```
s3='H'
print(s3.isupper())
```

```
True
```

In [42]:
```
s4='h'
print(s4.isupper())
```

```
False
```

In [43]:
```
s5="hello"
print(s5.islower())
```

```
True
```

In [46]:
```
s1="MOUNI"
k='+'.join(s1)
```

In [47]:
```
k
```

Out[47]: 'M+O+U+N+I'

In [50]:
```
s2="hai"
s3="welcome"
s4=s2.join(s3)
```

In [51]:
```
s4
```

Out[51]: 'whaiehailhaichaiohaimhaie'

In [52]:
```
s5=s2+s3
s5
```

Out[52]: 'haiwelcome'

In [53]:
```
s1="hello"
s1.swapcase()
```

Out[53]: 'HELLO'

In [55]:
```
s2="NI"
s2.swapcase()
```

Out[55]: 'ni'

In [56]:
```
s11="DIET college of Engineering"
s11.count('l')
```

Out[56]: 2

In [57]:
```
s11.index('c')
```

Out[57]: 5

In [61]:
```
s12="anakapalli"
s12.title()
```

Out[61]: 'Anakapalli'

In [62]:
```
s13="mounika cse second year"
s13.title()
```

Out[62]: 'Mounika Cse Second Year'

In [65]:
```
h='mounika'
print(h.endswith('a'))
print(h.startswith('m'))
```

```
True
True
```

In [66]:
```
h1="hello world"
h1.split('l')
```

Out[66]:
```
['he', '', 'o wor', 'd']
```

In [67]:
```
h3="good"
h3.replace('o','g')
```

Out[67]:
```
'gggd'
```

In [84]:
```
h3
```

Out[84]:
```
'good'
```

In [69]:
```
h4="12346"
h4.isnumeric()
```

Out[69]:
```
True
```

In [70]:
```
h4.isalnum()
```

Out[70]:
```
True
```

In [71]:
```
h5="123ef"
h5.isalnum()
```

Out[71]:
```
True
```

In [73]:
```
h6="adsgdjdl"
h6.isalpha()
```

Out[73]:
```
True
```

In [74]:
```
h7="mounika "
h7.split()
```

Out[74]:
```
['mounika']
```

In [83]:
```
 h="   mouni   "
print(len(h))
x=h.lstrip()
print(len(x))
y=h.rstrip()
print(len(y))
h.strip()
```

```
11
8
8
```

Out[83]:
```
'mouni'
```

conditional statements

- if
- else
- elseif

syntax

- if condition
  - statement
- else
  - statement

In [78]:
```
s="hello dite"
if type(s)==str:
    print('this is string')
else:
    print('not a string')
```

```
this is string
```

In [79]:
```
n=input()
print(type(n))
```

```
6
<class 'str'>
```

In [82]:
```
n=int(input("enter a number"))
if n%2==0:
     print("even number")
else:
     print("odd number")
```

```
enter a number5
odd number
```

In [85]:
```
n=int(input("enter your marks"))
if n>35:
    print("pass")
else:
    print("fail")
```

```
enter your marks99
pass
```

In [88]:
```
a=int(input())
b=int(input())
c=int(input())
if a>b and a>c:
        print('a is greatest')
elif b>c:
        print('b is greatest')
else:
        print('c is greatest')
```

```
1
2
3
c is greatest
```

In [91]:
```
year=int(input())
if year%400==0 or (year%4==0 and year%100!=0):
    print(366*24*60*60)
else:
    print(365)
```

```
2000
31622400
```

loops

In [7]:
```
i=0
while i<10:
       print(i,end=' ')
       i=i+1
```

```
0 1 2 3 4 5 6 7 8 9
```

In [10]:
```
i=0
while i<10:
    if i%2==0:
            print(i,end='')
    i=i+1
```

```
02468
```

In [2]:
```
i=0
while i<10:
    if i%2==1:
            print(i,end='')
    i=i+1
```

```
13579
```

while loop with break

In [1]:
```
i=0
while i<10:
    if i==7:
        break
    print(i,end='')
    i=i+1
```

```
0123456
```

In [1]:
```
i=0
while i<10:
    if i==7:
        pass
    print(i,end='')
    i=i+1
```

```
0123456789
```

In [2]:
```
i=0
while i<10:
    if i==7:
        break
    print(i)
    i=i+1
```

```
0
1
2
3
4
5
6
```

In [5]:
```
for i in range(10):
    if i==7:
        continue
    else:
         print(i)
```

```
0
1
2
3
4
5
6
8
9
```

In [10]:
```
for i in range(0,10):
      print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

In [11]:
```
a='abcdefgh'
for i in a:
    if i=='c':
        print(i)
```

```
c
```

In [16]:
```
for i in range(0,10+1):
      print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
```

In [17]:
```
for i in range(0,10+1):
      print(i)
```

```
0
1
2
3
4
5
6
7
8
9
10
```

In [18]:
```
for i in range(0,10+1):
    if i==5:
        break
    else:
       print(i)
```

```
0
1
2
3
4
```

In [20]:
```
pt='dad'
print(pt)
s2=pt[::-1]
print(s2)
if s2==pt:
    print('True')
else:
    print('False')
```

```
dad
dad
True
```

printing natural numberes

In [21]:
```
for i in range(0,10):
       print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

In [4]:
```
i=0
while i<10:
    print(i)
    i=i+1
```

```
0
1
2
3
4
5
6
7
8
9
```

alternative numbers

In [3]:
```
for i in range(0,10,2):
    print(i)
```

```
0
2
4
6
8
```

In [2]:
```
i=0
while i<10:
    print(i)
    i=i+2
```

```
0
2
4
6
8
```

In [ ]:
```
to print cube of number
```

In [6]:
```
n=int(input())
for i in range(n):
     print(i**3 ,end=' ')
```

```
11
0 1 8 27 64 125 216 343 512 729 1000
```

In [16]:
```
sum=0
n=int(input())
for i in range(n):
    sum=sum+i
print('sum of all numbers:',sum)
print('averge:',sum/n)
```

```
10
sum of all numbers: 45
averge: 4.5
```

In [17]:
```
sum=0
n=int(input())
for i in range(n):
    sum=sum+i
print(sum)
print(sum/n)
```

```
10
45
4.5
```

In [19]:
```
n=int(input())
s=0
for i in range(1,n):
    if n%i==0:
        s=s+i
if s==n:
       print("perfect number")
else:
        print("not a perfect")
```

```
6
perfect number
```

In [20]:
```
n=input()
digits=len(n)
total=0
for i in  n:
    total=total+int(i)**(digits)
if total==int(n):
    print('armstrong number')
else:
    print(' not an armstrong number')
```

```
153
armstrong number
```

function

types of function

- without arg&without return value
- without arg&with return value
- with arg&without return value
- with arg&with return value

In [22]:
```
#without arg&without return value
def add():
    a=10
    b=20
    c=a+b
    print(c)
add()
```

```
30
```

In [27]:
```
#with arg&without return value
def mul(a,b):
    c=a+b
    print(c)
mul(5,10)
```

```
15
```

In [28]:
```
# without arg&with return value
def sum1():
    a=10
    b=20
    sum=a+b
    return sum
print('calling outside fuction',sum1())
```

```
calling outside fuction 30
```

In [30]:
```
def add(a,b)
    return a+b
print(' outside fuction',add(1,2)
```

```
  File "<ipython-input-30-2670f1425b88>", line 1
    def add(a,b)
                ^
SyntaxError: invalid syntax
```

In [ ]:
```

```