set

```
s={}
s
```

```
{}
```

```
type(s)
```

```
dict
```

```
s1=set()
type(s1)
```

```
set
```

```
s1
```

```
set()
```

```
s2={20,100,3,45}
s2
```

```
{3, 20, 45, 100}
```

```
s3={'z','l','c','e','f'}
s3
```

```
{'c', 'e', 'f', 'l', 'z'}
```

```
s
```

```
{}
```

```
s4={1,2.3,'nit',1+2j,[1,2,3],(4,5,6),True}
s4
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[20], line 1
----> 1 s4={1,2.3,'nit',1+2j,[1,2,3],(4,5,6),True}
      2 s4

TypeError: unhashable type: 'list'
```

```
s5={2,3.4,'nit',1+2j,False}
s5
```

```
{(1+2j), 2, 3.4, False, 'nit'}
```

```
print(s1)
print(s2)
print(s3)
print(s5)
```

```
set()
{45, 3, 100, 20}
{'l', 'f', 'z', 'e', 'c'}
{False, 2, 3.4, (1+2j), 'nit'}
```

```
s2.add(30)
```

```
s2
```

```
{3, 20, 30, 45, 100}
```

```
s2.add(200)
s2
```

```
{3, 20, 30, 45, 100, 200}
```

```
s2[:]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[35], line 1
----> 1 s2[:]

TypeError: 'set' object is not subscriptable
```

```
s2
```

```
{3, 20, 30, 45, 100, 200}
```

```
s2[1:5]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[39], line 1
----> 1 s2[1:5]

TypeError: 'set' object is not subscriptable
```

```
s5
```

```
{(1+2j), 2, 3.4, False, 'nit'}
```

```
s4=s5.copy()
s4
```

```
{(1+2j), 2, 3.4, False, 'nit'}
```

```
s4.add(2)
s4
```

```
{(1+2j), 2, 3.4, False, 'nit'}
```

```
s5.clear()
```

```
s5
```

```
set()
```

```
del s5
```

```
s4
```

```
{(1+2j), 2, 3.4, False, 'nit'}
```

```
s4.remove((1+2j))
```

```
s4
```

```
{2, 3.4, False, 'nit'}
```

```
s3
```

```
{'c', 'e', 'f', 'l', 'z'}
```

```
s3.discard('m')
```

```
s3.remove('f')
```

```
s3
```

```
{'c', 'e', 'l', 'z'}
```

```
s3.discard('e')
s3
```

```
{'c', 'l', 'z'}
```

```
s3.pop()
```

```
'l'
```

```
s3
```

```
{'c', 'z'}
```

```
s2
```

```
{3, 20, 30, 45, 100, 200}
```

```
s2.pop()
```

```
3

s2

{20, 30, 45, 100, 200}

for i in s2:
    print(i)

100
200
45
20
30

s2

{20, 30, 45, 100, 200}

5 in s2

False

45 in s2

True

s2

{20, 30, 45, 100, 200}

s2.update(s3)

s2

{100, 20, 200, 30, 45, 'c', 'z'}

s3

{'c', 'z'}
```

set operations

```
s6={1,2,3,4,5}
s7={4,5,6,7,8}
s8={8,9,10}

s6.union(s7)

{1, 2, 3, 4, 5, 6, 7, 8}

s6.union(s7,s8)

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
s6|s7
```

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

```
s6|s7|s8
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
print(s6)
print(s7)
print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

```
s6.intersection(s7)
```

```
{4, 5}
```

```
s6.intersection(s8)
```

```
set()
```

```
s7.intersection(s8)
```

```
{8}
```

```
s6&s7
```

```
{4, 5}
```

```
s6&s7
```

```
{4, 5}
```

```
s7&s8
```

```
{8}
```

```
s6&s7&s8
```

```
set()
```

```
print(s6)
print(s7)
print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

```
s6.difference(s7)
```

```
{1, 2, 3}
```

```
s6-s7
```

```
{1, 2, 3}
```

```
s7-s8
```

```
{4, 5, 6, 7}
```

```
s6-s7-s8
```

```
{1, 2, 3}
```

```
s8-s7
```

```
{9, 10}
```

```
s8-s6
```

```
{8, 9, 10}
```

```
print(s6)
print(s7)
print(s8)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
{8, 9, 10}
```

```
s6.symmetric_difference(s7)
```

```
{1, 2, 3, 6, 7, 8}
```

```
s6.symmetric_difference(s8)
```

```
{1, 2, 3, 4, 5, 8, 9, 10}
```

```
s7.symmetric_difference(s6)
```

```
{1, 2, 3, 6, 7, 8}
```

```
s8.symmetric_difference(s7)
```

```
{4, 5, 6, 7, 9, 10}
```

```
s8.symmetric_difference(s6)
```

```
{1, 2, 3, 4, 5, 8, 9, 10}
```

```
print(s6)
```

```
{1, 2, 3, 4, 5}
```

```
print(s7)
```

```
{4, 5, 6, 7, 8}
```

```
print(s8)
```

```
{8, 9, 10}
```

```
myset={1,2,3,4,5}
myset
```

```
{1, 2, 3, 4, 5}
```

```
len(myset)
```

```
5
```

```
my_set={1,1,2,2,3,4,5,5}
my_set
```

```
{1, 2, 3, 4, 5}
```

```
myset1={1.79,2.08,3.99,4.56,5.45}
myset1
```

```
{1.79, 2.08, 3.99, 4.56, 5.45}
```

```
len(myset)
```

```
5
```

```
myset2={'asif','john','tyrion'}
myset2
```

```
{'asif', 'john', 'tyrion'}
```

```
myset3={10,20,'hola',(11,22,32)}
myset3
```

```
{(11, 22, 32), 10, 20, 'hola'}
```

```
myset3={10,20,'hola',[11,22,32]}
myset
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[151], line 1
----> 1 myset3={10,20,'hola',[11,22,32]}
      2 myset

TypeError: unhashable type: 'list'
```

```
myset4=set()
print(type(myset4))
```

```
<class 'set'>
```

```
my_set1=set(('one','two','three','four'))
my_set1
```

```
{'four', 'one', 'three', 'two'}
```

```
myset={'one','two','three','four','five','six','seven','eight'}
for i in myset:
    print(i)
```

```
three
six
one
seven
four
five
two
eight
```

```
for i in enumerate (myset):
    print(i)
```

```
(0, 'three')
(1, 'six')
(2, 'one')
(3, 'seven')
(4, 'four')
(5, 'five')
(6, 'two')
(7, 'eight')
```

```
myset
```

```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
'one'in myset
```

```
True
```

```
'ten' in myset
```

```
False
```

```
if 'three' in myset:
    print('three is present in the set')
else:
    print('three is not present in the set')
```

```
three is present in the set
```

```
if 'eleven' in myset:
    print('eleven is present in the set')
```

```
else:
    print('eleven is not present in the set')
```

```
eleven is not present in the set
```

```
myset
```

```
{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}
```

```
myset.add('nine')
myset
```

```
{'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three',
'two'}
```

```
myset.update(['ten','eleven','twelve'])
myset
```

```
{'eight',
 'eleven',
 'five',
 'four',
 'nine',
 'one',
 'seven',
 'six',
 'ten',
 'three',
 'twelve',
 'two'}
```

```
myset.remove('nine')
myset
```

```
{'eight',
 'eleven',
 'five',
 'four',
 'one',
 'seven',
 'six',
 'ten',
 'three',
 'twelve',
 'two'}
```

```
myset.discard('ten')
myset
```

```
{'eight',
 'eleven',
 'five',
```

```
 'four',
 'one',
 'seven',
 'six',
 'three',
 'twelve',
 'two'}

myset.clear()
myset

set()

del myset

myset

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[187], line 1
----> 1 myset

NameError: name 'myset' is not defined

myset={'one','two','three','four','five','six','seven','eight'}
myset

{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

myset1=myset
myset1

{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

id(myset),id(myset)

(2212281410368, 2212281410368)

my_set=myset.copy()
my_set

{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

id(my_set)

2212281416864

myset.add('nine')
myset
```

```
{'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three',
'two'}

myset1

{'eight', 'five', 'four', 'nine', 'one', 'seven', 'six', 'three',
'two'}

my_set

{'eight', 'five', 'four', 'one', 'seven', 'six', 'three', 'two'}

a={1,2,3,4,5}
b={4,5,6,7,8}
c={8,9,10}

a|b

{1, 2, 3, 4, 5, 6, 7, 8}

a.union(b)

{1, 2, 3, 4, 5, 6, 7, 8}

a.union(b,c)

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

a.update(b,c)
a

{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

a={1,2,3,4,5}
b={4,5,6,7,8}

a&b

{4, 5}

a.intersection(b)

{4, 5}

a.intersection_update(b)
a

{4, 5}

a={1,2,3,4,5}
b={4,5,6,7,8}

a-b
```

```
{1, 2, 3}
```

```
a.difference(b)
```

```
{1, 2, 3}
```

```
b-a
```

```
{6, 7, 8}
```

```
b.difference(a)
```

```
{6, 7, 8}
```

```
b.difference_update(a)
b
```

```
{6, 7, 8}
```

```
a={1,2,3,4,5}
b={4,5,6,7,8}
```

```
a^b
```

```
{1, 2, 3, 6, 7, 8}
```

```
a.symmetric_difference(b)
```

```
{1, 2, 3, 6, 7, 8}
```

```
a.symmetric_difference_update(b)
a
```

```
{1, 2, 3, 6, 7, 8}
```

```
a={1,2,3,4,5,6,7,8,9}
b={3,4,5,6,7,8}
c={10,20,30,40}
```

```
b.issubset(a)
```

```
True
```

```
a.issuperset(b)
```

```
True
```

```
c.isdisjoint(a)
```

```
True
```

```
b.isdisjoint(a)
```

```
False
```

```
a
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
sum(a)
```

```
45
```

```
max(a)
```

```
9
```

```
min(a)
```

```
1
```

```
len(a)
```

```
9
```

```
list(enumerate(a))
```

```
[(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8,
9)]
```

```
d=sorted(a,reverse=True)
d
```

```
[9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
sorted(d)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

10 th mar

superset subset disjoint

```
s11={1,2,3,4,5,6,7,8,9}
s12={3,4,5,6,7,8}
s13={10,20,30,40}
```

```
s12.issubset(s11)
```

```
True
```

```
s11.issubset(s12)
```

```
False
```

```
s11.issuperset(s12)
```

```
True
```

```
s13.isdisjoint(s12)
```

```
True
```

```
s13.isdisjoint(s11)
```

```
True
```

```
s12={1,2,3,4,5}
s13={10,20,30}
s14={15.25,35}
```

```
s13.issubset(s12)
```

```
False
```

```
s12.issuperset(s13)
```

```
False
```

```
s14.isdisjoint(s12)
```

```
True
```

```
s14.isdisjoint(s13)
```

```
True
```

```
s13.isdisjoint(s12)
```

```
True
```

```
s12.isdisjoint(s14)
```

```
True
```

```
s12.issuperset(s13)
```

```
False
```

```
s13.issuperset(s14)
```

```
False
```

```
s15={1,2,3,4,5,6}
s16={6,7,8,9}
s17={10,20}
```

```
s16.issubset(s15)
```

```
False
```

```
s17.isdisjoint(s16)
```

```
True
```

```
s14
```

```
{15.25, 35}
```

```
s15
```

```
{1, 2, 3, 4, 5, 6}
```

```
for i in s15:
    print(i)
```

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

```
for i in enumerate(s15):
    print(i)
```

```
(0, 1)
(1, 2)
(2, 3)
(3, 4)
(4, 5)
(5, 6)
```

```
s15
```

```
{1, 2, 3, 4, 5, 6}
```

```
sum(s15)
```

```
21
```

```
min(s15)
```

```
1
```

```
max(s15)
```

```
6
```

dictionary

```
d={}
d
```

```
{}
```

```
type(d)
```

```
dict
```

```
d1={1:'one',2:'two',3:'three'}
```

```
d1
```

```
{1: 'one', 2: 'two', 3: 'three'}
```

```
d1.keys()
```

```
dict_keys([1, 2, 3])
```

```
d1.values()
```

```
dict_values(['one', 'two', 'three'])
```

```
d2=d1.copy()
d2
```

```
{1: 'one', 2: 'two', 3: 'three'}
```

```
d1.items()
```

```
dict_items([(1, 'one'), (2, 'two'), (3, 'three')])
```

```
d1[1]
```

```
'one'
```

```
keys={'ran','b','c','d'}
value=[10,20,30]
mydict3=dict.fromkeys(keys,value)
mydict3
```

```
{'b': [10, 20, 30], 'ran': [10, 20, 30], 'd': [10, 20, 30], 'c': [10,
20, 30]}
```

```
value.append(50)
mydict3
```

```
{'b': [10, 20, 30, 50],
 'ran': [10, 20, 30, 50],
 'd': [10, 20, 30, 50],
 'c': [10, 20, 30, 50]}
```

```
range(10)
```

```
range(0, 10)
```

```
list(range(0,10))
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
list(range(10,20,3))
```

```
[10, 13, 16, 19]
```

```
list(range(10,20,4))
```

```
[10, 14, 18]
```

```
r=range(1,10)
r
```

```
range(1, 10)
```

```
for i in r:
    print(i)
```

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

```
mydict=dict()
mydict
```

```
{}
```

```
mydict={}
mydict
```

```
{}
```

```
mydict={1:'one',2:'two',3:'three'}
mydict
```

```
{1: 'one', 2: 'two', 3: 'three'}
```

```
mydict=dict({1:'one',2:'two',3:'three'})
mydict
```

```
{1: 'one', 2: 'two', 3: 'three'}
```

```
mydict={'A':'one','B':'two', 'C':'three'}
mydict
```

```
{'A': 'one', 'B': 'two', 'C': 'three'}
```

```
mydict={1:'one','A':'two',3:'three'}
mydict
```

```
{1: 'one', 'A': 'two', 3: 'three'}
```

```
mydict.keys()
```

```
dict_keys([1, 'A', 3])
```

```
mydict.values()
```

```
dict_values(['one', 'two', 'three'])
```

```
mydict.items()
```

```
dict_items([(1, 'one'), ('A', 'two'), (3, 'three')])
```

```
mydict={1:'one',2:'two',3:'three'}
mydict
```

```
{1: 'one', 2: 'two', 3: 'three'}
```

```
mydict={1:'one',2:'two','A':['asif','john','maria']}
mydict
```

```
{1: 'one', 2: 'two', 'A': ['asif', 'john', 'maria']}
```

```
mydict={1:'one',2:'two','A':['asif','john','maria'],'B':('bat','cat')}
mydict
```

```
{1: 'one', 2: 'two', 'A': ['asif', 'john', 'maria'], 'B': ('bat',
'cat')}
```

```
keys={'a','b','c','d'}
mydict3=dict.fromkeys(keys)
mydict3
```

```
{'b': None, 'a': None, 'd': None, 'c': None}
```

```
keys={'a','b','c','d'}
value=10
mydict3=dict.fromkeys(keys,value)
mydict3
```

```
{'b': 10, 'a': 10, 'd': 10, 'c': 10}
```

```
keys={'a','b','c','d'}
value=[10,20,30]
mydict3=dict.fromkeys(keys,value)
mydict3
```

```
{'b': [10, 20, 30], 'a': [10, 20, 30], 'd': [10, 20, 30], 'c': [10,
20, 30]}
```

```
value.append(40)
mydict3
```

```
{'b': [10, 20, 30, 40],
 'a': [10, 20, 30, 40],
 'd': [10, 20, 30, 40],
 'c': [10, 20, 30, 40]}
```

```
mydict={1:'one',2:'two',3:'three',4:'four'}
mydict
```

```
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
```

```
mydict[1]
```

```
'one'
```

```
mydict.get(1)
```

```
'one'
```

```
mydict1={'name':'asif','id':74123,'dob':1991,'job':'analyst'}
mydict1
```

```
{'name': 'asif', 'id': 74123, 'dob': 1991, 'job': 'analyst'}
```

```
mydict1['name']
```

```
'asif'
```

```
mydict1.get('job')
```

```
'analyst'
```

```
mydict1={'name':'asif','id':12345,'dob':1991,'address':'hilsinki'}
mydict1
```

```
{'name': 'asif', 'id': 12345, 'dob': 1991, 'address': 'hilsinki'}
```

```
mydict1['dob']=1992
mydict['address']='delhi'
mydict1
```

```
{'name': 'asif', 'id': 12345, 'dob': 1992, 'address': 'hilsinki'}
```

```
dict1={'dob':1995}
mydict1.update(dict1)
mydict1
```

```
{'name': 'asif', 'id': 12345, 'dob': 1995, 'address': 'hilsinki'}
```

```
mydict1['job']='analyst'
mydict1
```

```
{'name': 'asif',
 'id': 12345,
 'dob': 1995,
```

```
 'address': 'hilsinki',
 'job': 'analyst'}

mydict1.pop('job')
mydict1

{'name': 'asif', 'id': 12345, 'dob': 1995, 'address': 'hilsinki'}

mydict1.popitem()

('address', 'hilsinki')

mydict1

{'name': 'asif', 'id': 12345, 'dob': 1995}

del[mydict1['id']]

mydict1.clear()
mydict1

{}

del mydict1
mydict1

---------------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[85], line 2
      1 del mydict1
----> 2 mydict1

NameError: name 'mydict1' is not defined

mydict={'name':'asif','id':12345,'dob':1991,'address':'hilsinki'}
mydict

{'name': 'asif', 'id': 12345, 'dob': 1991, 'address': 'hilsinki'}

mydict1=mydict

id(mydict),id(mydict1)

(2212294092992, 2212294092992)

mydict2=mydict.copy()

id(mydict2)

2212294069248

mydict['address']='mumbai'
```

```
mydict
```

```
{'name': 'asif', 'id': 12345, 'dob': 1991, 'address': 'mumbai'}
```

```
mydict1
```

```
{'name': 'asif', 'id': 12345, 'dob': 1991, 'address': 'mumbai'}
```

```
mydict2
```

```
{'name': 'asif', 'id': 12345, 'dob': 1991, 'address': 'hilsinki'}
```

```
mydict1={'name':'asif','id':12345,'dob':1991,'address':'hilsinki','job
':'analyst'}
mydict1
```

```
{'name': 'asif',
 'id': 12345,
 'dob': 1991,
 'address': 'hilsinki',
 'job': 'analyst'}
```

```
for i in mydict1:
    print(i,':',mydict1[i])
```

```
name : asif
id : 12345
dob : 1991
address : hilsinki
job : analyst
```

```
for i in mydict1:
    print(mydict1[i])
```

```
asif
12345
1991
hilsinki
analyst
```

```
mydict1={'name':'asif','id':12345,'dob':1991,'job':'analyst'}
mydict1
```

```
{'name': 'asif', 'id': 12345, 'dob': 1991, 'job': 'analyst'}
```

```
'name' in mydict1
```

```
True
```

```
'asif' in mydict1
```

```
False
```

```
'id' in mydict1
```

```
True
```

```
'address' in mydict1
```

```
False
```

```
mydict1={'name':'asif','id':12345,'dob':1991,'job':'analyst'}
mydict1
```

```
{'name': 'asif', 'id': 12345, 'dob': 1991, 'job': 'analyst'}
```

```
all(mydict1)
```

```
True
```

manipulating strings

```
print('hello there!\n how are you?\n i\'m doing fine.')
```

```
hello there!
 how are you?
 i'm doing fine.
```

```
>>> print(r"Hello there!\nHow are you?\nI\'m doing fine.")
```

```
Hello there!\nHow are you?\nI\'m doing fine.
```

```
print(
... """Dear Alice,
...
... Eve's cat has been arrested for catnapping,
... cat burglary, and extortion.
...
... Sincerely,
... Bob"""
... )
```

```
Dear Alice,

Eve's cat has been arrested for catnapping,
cat burglary, and extortion.

Sincerely,
Bob
```

```
spam = 'Hello world!'
```

```
spam[0]
```

```
'H'
```

```
spam[4]
'o'
spam[-1]
'!'
spam
'Hello world!'
spam[0:5]
'Hello'
spam[:5]
'Hello'
spam[6:-1]
'world'
spam[:-1]
'Hello world'
spam[::-1]
'!dlrow olleH'
fizz=spam[0:5]
fizz
'Hello'
'hello' in 'hello world'
True
'Hello' in 'Hello'
True
'HELLO' in 'Hello World'
False
'' in 'spam'
True
'cats' not in 'cats and dogs'
```

```
False
```

```
greet = 'Hello world!'
greet.upper()
```

```
'HELLO WORLD!'
```

```
greet.lower()
```

```
'hello world!'
```

```
greet.title()
```

```
'Hello World!'
```

```
spam = 'Hello world!'
spam.islower()
```

```
False
```

```
spam.isupper()
```

```
False
```

```
'HELLO'.isupper()
```

```
True
```

```
'abc12345'.islower()
```

```
True
```

```
'12345'.islower()
```

```
False
```

```
'12345'.isupper()
```

```
False
```

```
'Hello world!'.startswith('Hello')
```

```
True
```

```
'Hello world!'.endswith('world!')
```

```
True
```

```
'abc123'.startswith('abcdef')
```

```
False
```

```
'abc123'.endswith('12')
```

```
False
```

```
'Hello world!'.startswith('Hello world!')
True
'Hello world!'.endswith('Hello world!')
True
''.join(['My', 'name', 'is', 'Simon'])
'MynameisSimon'
', '.join(['cats', 'rats', 'bats'])
'cats, rats, bats'
' '.join(['My', 'name', 'is', 'Simon'])
'My name is Simon'
'ABC'.join(['My', 'name', 'is', 'Simon'])
'MyABCnameABCisABCSimon'
'My name is Simon'.split()
['My', 'name', 'is', 'Simon']
'MyABCnameABCisABCSimon'.split('ABC')
['My', 'name', 'is', 'Simon']
'My name is Simon'.split('m')
['My na', 'e is Si', 'on']
'My name is Simon'.split()
['My', 'name', 'is', 'Simon']
'My name is Simon'.split(' ')
['My', 'name', 'is', 'Simon']
'HELLO'.rjust(10)
'     HELLO'
'HELLO'.rjust(20)
'               HELLO'
'HELLO'.rjust(10)
'HELLO world'.rjust(10)
```

```
'HELLO world'
```

```
'HELLO'.ljust(10)
```

```
'HELLO     '
```

```
'HELLO'.center(10)
```

```
'  HELLO   '
```

```
'Hello'.rjust(20, '*')
```

```
'***************Hello'
```

```
'Hello'.ljust(20, '-')
```

```
'Hello---------------'
```

```
'Hello'.center(20, '=')
```

```
'=======Hello========'
```

```
spam = ' Hello World '
spam.strip()
```

```
'Hello World'
```

```
spam.lstrip()
```

```
'Hello World '
```

```
spam.rstrip()
```

```
' Hello World'
```

```
spam = 'SpamSpamBaconSpamEggsSpamSpam'
spam.strip('ampS')
```

```
'BaconSpamEggs'
```

```
sentence = 'one sheep two sheep three sheep four'
sentence.count('sheep')
```

```
3
```

```
sentence.count('e')
```

```
9
```

```
sentence.count('e',6)
```

```
8
```

```
sentence.count('e',7)
```

```
7
```

```
text = "Hello, world!"
text.replace("world" ,"planet")
```

```
'Hello, planet!'
```

```
fruits = "apple, banana, cherry, apple"
fruits.replace("apple", "orange", 1)
```

```
'orange, banana, cherry, apple'
```

```
sentence = "I like apples, Apples are my favorite fruit"
sentence.replace("apples", "oranges")
```

```
'I like oranges, Apples are my favorite fruit'
```