Formatowanie stringów

```
a,b=1/3,'1/3'
'a=%f,  b=%s'%(a,b)                             #0.209
'a=%.3f,  b=%s'%(a,b)                           #0.231

'a={},  b={}'.format(a,b)                       #0.601
'a={:.3f},  b={}'.format(a,b)                   #0.274

f'a={a},  b={b}'                                #0.539
f'a={a:.3f},  b={b}'                            #0.224

f'{a=},  {b=}'                                  #0.523
f'{a=:.3f},  {b=}'                              #0.259
```

Formatowanie stringów

```
a,b=1/3,'1/3'
'a=%f, b=%s'%(a,b)                          #0.209
'a=%.3f, b=%s'%(a,b)                        #0.231

'a={}, b={}'.format(a,b)                    #0.601
'a={:.3f}, b={}'.format(a,b)                #0.274

f'a={a}, b={b}'                             #0.539
f'a={a:.3f}, b={b}'                         #0.224

f'{a=}, {b=}'                               #0.523
f'{a=:.3f}, {b=}'                           #0.259


a,b=3,'1/3'
'a=%f, b=%s'%(a,b)                          #0.142

'a={}, b={}'.format(a,b)                    #0.179

f'a={a}, b={b}'                             #0.111

f'{a=}, {b=}'                               #0.142
```

Stringi

```
print('abcd efgh'.capitalize())                 #Abcd efgh
print('abcd efgh'.title())                      #Abcd Efgh

print('abcdEFGH'.lower())                       #abcdefgh
print('abcdEFGH'.swapcase())                    #ABCDefgh
print('abcdEFGH'.upper())                       #ABCDEFGH

print('abcdefgh'.center(20))                    #      abcdefgh
print('abcdefgh'.center(20,'*'))                #******abcdefgh******
print('abcdefgh'.ljust(20))                     #abcdefgh
print('abcdefgh'.ljust(20,'*'))                 #abcdefgh************
print('abcdefgh'.rjust(20))                     #            abcdefgh
print('abcdefgh'.rjust(20,'*'))                 #************abcdefgh

print('abcdefgh'.removeprefix('ab'))            #cdefgh
print('abcdefgh'.removesuffix('gh'))            #abcdef

print('abcdefgh'.zfill(20))                     #000000000000abcdefgh

print('    abcdefgh    '.lstrip()+'*')          #abcdefgh    *
print('    abcdefgh    '.rstrip()+'*')          #    abcdefgh*
print('    abcdefgh    '.strip()+'*')           #abcdefgh*

print('abcd    efgh'.count(' '))                #0
print('abcd    efgh'.expandtabs().count(' '))   #4
```

Stringi

```
print('{0}abcdefgh{0}'.format('*'))                  #*abcdefgh*
print('{1}abcdefgh{0}'.format('*', '**'))            #**abcdefgh*
print('{s}abcdefgh{s}'.format(s='*'))                #*abcdefgh*
print('{s2}abcdefgh{s}'.format(s='*', s2='**'))      #**abcdefgh*
print('{s2}abcdefgh{0}'.format('*', s2='**'))        #**abcdefgh*
print('{}abcdefgh{}'.format('*', '**'))              #*abcdefgh**
```

Stringi

```
print('{0}abcdefgh{0}'.format('*'))             #*abcdefgh*

print('{1}abcdefgh{0}'.format('*', '**'))       #**abcdefgh*

print('{s}abcdefgh{s}'.format(s='*'))           #*abcdefgh*

print('{s2}abcdefgh{s}'.format(s='*', s2='**')) #**abcdefgh*

print('{s2}abcdefgh{0}'.format('*', s2='**'))   #**abcdefgh*

print('{}abcdefgh{}'.format('*', '**'))         #*abcdefgh**


class Foo(dict):
 def __missing__(self, key):
  return key

print('{s3}abcdefgh{s}'.format_map(Foo(s='*', s2='**')))
                                                #s3abcdefgh*
```

Stringi

- s.isalnum(), s.isalpha(), s.isascii()
- s.isdecimal(), s.isdigit(), s.isnumeric()
- s.isidentifier()
  keyword.iskeyword(s)
- s.islower(), s.istitle(), s.isupper()
- s.isprintable(), s.isspace()
- s.casefold()

Stringi

```
print('abrakadabra'.count('a'))              #5

print('abrakadabra'.startswith('ab'))        #True
print('abrakadabra'.endswith('a'))           #True

print('abrakadabra'.find('a'))               #0
print('abrakadabra'.find('z'))               #-1
print('abrakadabra'.find('a',4))             #5
print('abrakadabra'.rfind('a'))              #10

print('abrakadabra'.index('a'))              #0
print('abrakadabra'.index('z'))              #wyjątek
print('abrakadabra'.index('a',4))            #5
print('abrakadabra'.rindex('a'))             #10
```

Stringi

```
print('abra kadabra'.split())             #['abra', 'kadabra']
print('abra kadabra'.rsplit())            #['abra', 'kadabra']
```

Stringi

```
print('abra kadabra'.split())           #['abra', 'kadabra']
print('abra kadabra'.rsplit())          #['abra', 'kadabra']


print('abra kadabra'.split('r'))        #['ab', 'a kadab', 'a']
print('abra kadabra'.rsplit('r'))       #['ab', 'a kadab', 'a']

print('abra kadabra'.split('r',1))      #['ab', 'a kadabra']
print('abra kadabra'.rsplit('r',1))     #['abra kadab', 'a']
```

Stringi

```python
print('abra kadabra'.split())           #['abra', 'kadabra']
print('abra kadabra'.rsplit())          #['abra', 'kadabra']



print('abra kadabra'.split('r'))        #['ab', 'a kadab', 'a']
print('abra kadabra'.rsplit('r'))       #['ab', 'a kadab', 'a']


print('abra kadabra'.split('r',1))      #['ab', 'a kadabra']
print('abra kadabra'.rsplit('r',1))     #['abra kadab', 'a']

import re
print(re.split('[rb]','abra kadabra'))
                                  #['a', '', 'a kada', '', 'a']
```

Stringi

```
print('abra kadabra'.split())           #['abra', 'kadabra']
print('abra kadabra'.rsplit())          #['abra', 'kadabra']



print('abra kadabra'.split('r'))        #['ab', 'a kadab', 'a']
print('abra kadabra'.rsplit('r'))       #['ab', 'a kadab', 'a']


print('abra kadabra'.split('r',1))      #['ab', 'a kadabra']
print('abra kadabra'.rsplit('r',1))     #['abra kadab', 'a']

import re
print(re.split('[rb]','abra kadabra'))
                                #['a', '', 'a kada', '', 'a']


print('abra\nkadabra'.splitlines())     #['abra', 'kadabra']
```

Stringi

```
print('abra kadabra'.split())              #['abra', 'kadabra']
print('abra kadabra'.rsplit())             #['abra', 'kadabra']



print('abra kadabra'.split('r'))           #['ab', 'a kadab', 'a']
print('abra kadabra'.rsplit('r'))          #['ab', 'a kadab', 'a']


print('abra kadabra'.split('r',1))         #['ab', 'a kadabra']
print('abra kadabra'.rsplit('r',1))        #['abra kadab', 'a']

import re
print(re.split('[rb]','abra kadabra'))
                                     #['a', '', 'a kada', '', 'a']


print('abra\nkadabra'.splitlines())        #['abra', 'kadabra']



print('abrakadabra'.partition('br'))       #('a', 'br', 'akadabra')
print('abrakadabra'.rpartition('br'))      #('abrakada', 'br', 'a')
```

Stringi

```
print(''.join(('a','b','c')))                #abc
print('*'.join(('a','b','c')))               #a*b*c
```

Stringi

```
print(''.join(('a','b','c')))                  #abc
print('*'.join(('a','b','c')))                 #a*b*c

print('abrakadabra'.replace('a','A'))          #AbrAkAdAbrA
```

Stringi

```
print(''.join(('a','b','c')))                  #abc
print('*'.join(('a','b','c')))                 #a*b*c


print('abrakadabra'.replace('a','A'))          #AbrAkAdAbrA


tr=str.maketrans('abc','ABC')
print(tr)                                      #{97: 65, 98: 66, 99: 67}
print('abrakadabra'.translate(tr))             #ABrAkAdABrA
```

Stringi

```
print(''.join(('a','b','c')))              #abc
print('*'.join(('a','b','c')))             #a*b*c


print('abrakadabra'.replace('a','A'))      #AbrAkAdAbrA


tr=str.maketrans('abc','ABC')
print(tr)                                  #{97: 65, 98: 66, 99: 67}
print('abrakadabra'.translate(tr))         #ABrAkAdABrA


tr=str.maketrans('abc','ABC','r')
print(tr)                       #{97: 65, 98: 66, 99: 67, 114: None}
print('abrakadabra'.translate(tr))         #ABAkAdABA
```

Moduł string

- ascii_letters
- ascii_lowercase
- ascii_uppercase
- capwords
- digits
- hexdigits
- octdigits
- printable
- punctuation
- whitespace