

## Examples of String Operations: Screen grabs from idle session

```
Python 2.7.10 Shell
Python 2.7.10 (default, Jul 14 2015, 19:46:27)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

>>> nickname1 = "peanuts"
>>> nickname2 = 'bergerbits'
>>> allmynames = """
peanuts
bergerbits
skoshiwosh
zany
"""
>>> allmynames
'\npeanuts\nbergerbits\nskoshiwosh\nzany\n'
>>> print(allmynames)

peanuts
bergerbits
skoshiwosh
zany

>>> nickname1
'peanuts'
>>> nickname2
'bergerbits'
>>> # you can add strings together like numbers
>>> nickname1 + ' ' + nickname2
'peanuts bergerbits'
>>> # but preferable to use % or format for strings
>>> "My nicknames are %s and %s" % (nickname1, nickname2)
'My nicknames are peanuts and bergerbits'
>>> # above could be contained in print function
>>> print("My nicknames are %s and %s" % (nickname1, nickname2))
My nicknames are peanuts and bergerbits
>>> # use 'r' for raw string: good if escape character in string
>>> foobar = r'foo\noo'
>>> foobar
'foo\noo'
>>> # raw strings good for Windows file paths due to backslash in Windows path
>>>
>>> # new style string format
>>> print("My nicknames are {0} and {1}".format(nickname1, nickname2))
My nicknames are peanuts and bergerbits
>>> nickname1[0]
'p'
>>> nickname1[-1]
's'
>>> nickname[3:5]

Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    nickname[3:5]
NameError: name 'nickname' is not defined
>>> nickname1[3:5]
'nu'
>>> nickname1[0:-1:2]
'pau'
>>> |
```

```

>>> # strings are immutable so you can't do this:
>>> nickname1[4] = 'x'

Traceback (most recent call last):
  File "<pyshell#32>", line 1, in <module>
    nickname1[4] = 'x'
TypeError: 'str' object does not support item assignment
>>> # to change a string variable you have to re-assign it
>>> nickname3 = nickname1[0:4] + 'x' + nickname1[4:]
>>> nickname3
'peanxuts'
>>> # here are some common string methods
>>> # note that results must be re-assigned to string variable
>>> nickname1.upper()
'PEANUTS'
>>> nickname1.capitalize()
'Peanuts'
>>> nickname3.index('x')
4
>>> nickname3.index('y')

Traceback (most recent call last):
  File "<pyshell#41>", line 1, in <module>
    nickname3.index('y')
ValueError: substring not found
>>> nickname3.find('y')
-1
>>> nickname3.find('x')
4
>>> if nickname3.find('x') >= 0:
    print("found")

found
>>> foobar = "  foobar  "
>>> foobar.strip()
'foobar'
>>> foobar = "*foobar*"
>>> foobar.strip('*')
'foobar'
>>> foobar.rstrip('*')
'*foobar'
>>> all_names = "{} {} {} skoshiwosh zany".format(nickname1, nickname2, nickname3)
>>> all_names
'peanuts bergerbits peanxuts skoshiwosh zany'
>>> name_list = all_names.split()
>>> name_list
['peanuts', 'bergerbits', 'peanxuts', 'skoshiwosh', 'zany']
>>> new_all_names = '***'.join(name_list)
>>> new_all_names
'peanuts***bergerbits***peanxuts***skoshiwosh***zany'
>>> new_all_names.startswith('p')
True
>>> new_all_names.isdigit()
False
>>> new_all_names.isalpha()
False
>>> nickname1.isalpha()
True

```