

# Strings

In [4]:



```
# Declare Strings using '' OR ""  
# Declare Empty String  
a = ''  
print(a)  
b = ""  
print(b)  
c = str()  
print(c)  
print(type(a))
```

<class 'str'>

## String Index(Left to Right and Right to Left) ¶

- H e l l o
- 0 1 2 3 4 --> Left to Right
- -5 -4 -3 -2 -1 --> Right to Left

In [9]:



```
name = 'narayana'  
print(name)  
name  
thing = "car"  
thing
```

narayana

Out[9]:

'car'

In [14]:



```
# Index syntax for print single index -->variblename[index]  
print(name[0])  
print(name[-8])  
#print(name[10])
```

n  
n

In [19]:



```
# Concatination (+)
print(name + thing)
#print(name+123) # Wrong Concatinate
print(name+'hello')
```

```
narayanacar
narayanahello
```

In [22]:



```
# Duplication (*)
(name+" ")*5
```

Out[22]:

```
'narayana narayana narayana narayana narayana '
```

In [23]:



```
"hi"*2/2
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-23-cb53cc02a127> in <module>
----> 1 "hi"*2/2
```

**TypeError:** unsupported operand type(s) for /: 'str' and 'int'

In [25]:



```
"hi"*(2//2)
```

Out[25]:

```
'hi'
```

In [26]:



```
"hello"*int(2/2)
```

Out[26]:

```
'hello'
```

In [32]:



```
# Slicing (:)
# [intialize:condition:increment/Decrement]
# min 1 Arg -->[intial]
# 2 Arg      -->[intial : condition]
# max 3 Arg  -->[intial : cond : incre/decre]
temp = "python programming"
print(temp[1:])
print(temp[2:5])# n-1
print(temp[1:6:2])# increment
print(temp[5:1:-1])# decrement
print(temp[5:2:1])
print(temp[0:500]) # slicing not returns index out of range Errors
```

ython programming  
tho  
yhn  
noht

python programming

In [35]:



```
# Overcome Escape Sequence
path = "C:\\Users\\NARAYANA\\Desktop\\"
path = "C:/Users/NARAYANA/Desktop/"
path = r"C:\Users\NARAYANA\Desktop" # r means raw path
print(path)
```

C:\Users\NARAYANA\Desktop

In [38]:



```
# String Methods
#print(dir(path))
#print(dir("apssdc"))
print(dir(str))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
 '__eq__', '__format__', '__ge__', '__getattr__', '__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 [39]:



```
# Syntax of method
# variblename.methodname(Arg1,Arg2,...)
# variablevalue.methodname(Arg1,Arg2,....)
```

In [40]:



```
# String is a immutable(not possible update/delete) data type()
```

In [45]:



```
print("apssdc".upper())
print("NAME".upper())
print("coding".lower())
print('DON'.lower())
print(('monday'.upper()).lower())
names ="ramu srinu sana kiran"
print(names.upper())
```

APSSDC  
NAME  
coding  
don  
monday  
RAMU SRINU SANA KIRAN

In [49]:



```
# isupper()
print(names.isupper())
print((names.upper()).isupper())
print('laptop'.isupper())
```

False  
True  
False

In [50]:



```
# islower()
print(names.islower())
```

True

In [51]:



```
# replace()
names.replace('sana','ramya')
```

Out[51]:

'ramu srinu ramya kiran'

In [54]:



```
"grammer".replace('q','z')
```

Out[54]:

```
'grammer'
```

In [60]:



```
# split() #split returns List O/P
print(names)
print(names.split())
print("ramu is a doctor".split('o'))
print("narayana is a good boy".split('a'))
print("narayana is a good boy".split('a',2))
print("narayana is a good boy".split('a',100))
```

```
ramu srinu sana kiran
['ramu', 'srinu', 'sana', 'kiran']
['ramu is a d', 'ct', 'r']
['n', 'r', 'y', 'n', ' is ', ' good boy']
['n', 'r', 'yana is a good boy']
['n', 'r', 'y', 'n', ' is ', ' good boy']
```

In [61]:



```
# rsplit() #Right split returns List O/P
print(names.rsplit())
print("ramu is a doctor".rsplit('o'))
print("narayana is a good boy".rsplit('a'))
print("narayana is a good boy".rsplit('a',2))
print("narayana is a good boy".rsplit('a',100))
```

```
['ramu', 'srinu', 'sana', 'kiran']
['ramu is a d', 'ct', 'r']
['n', 'r', 'y', 'n', ' is ', ' good boy']
['narayan', ' is ', ' good boy']
['n', 'r', 'y', 'n', ' is ', ' good boy']
```

In [67]:



```
# find()
print("naveen".find('v'))
print("naveen".find('e'))
print("naveen".rfind('e')) # Right find
print('vishnu'.find('z'))
```

```
2
3
4
-1
```

In [69]:



```
# index()
print("shiva".index('h'))
print("shiva".index('x'))
```

1

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-69-1f506c133959> in <module>
      1 # index()
      2 print("shiva".index('h'))
----> 3 print("shiva".index('x'))
```

**ValueError:** substring not found

In [71]:



```
# Declare Multi Line String
paragraph = """first line
second line
trird line
...
..
"""
a1 = '''
one
two
three
'''
print(paragraph)
print(a1)
```

```
first line
second line
trird line
...
..
```

```
one
two
three
```

In [73]:



```
# splitlines()
print(a1.splitlines())
print(paragraph.splitlines())
```

```
['', 'one', 'two', 'three']
['first line', 'second line ', 'trird line', '...', '..']
```

In [75]:



```
# isdigit()
print("name122ravi".isdigit())
print("satya1"[-1].isdigit())
```

False

True

In [76]:



```
a="abc" # single line string
b = """aaa #multi line string
bbb
ccc
ddd
"""

print(a[1])
print("xyz"[-1])
```

b

z

In [79]:



```
a="abc \n bbb" # single line string
b = """aaa #multi line string
bbb
ccc
ddd
"""

print(a.split())
print(b.split())
```

['abc', 'bbb']

['aaa', '#multi', 'line', 'string', 'bbb', 'ccc', 'ddd']

In [82]:



```
digits ="1 2 3 4 5 6 7"
print(digits.split())
digits ="1,2,3,4,5,6,7"
print(digits.split(','))
```

['1', '2', '3', '4', '5', '6', '7']

['1', '2', '3', '4', '5', '6', '7']

In [ ]:

