Strings

```
# String assignment
text = "san francisco"
type(text)
```

```
str
```

```
num = ["24","34","36"]
type(num)
```

```
list
```

String Manipulations

```
var3 = "Var1" + "-" + "Var2"
var3
```

```
'Var1-Var2'
```

```
from itertools import cycle

l1 = [1, 2, 3, 4, 5]
l2 = ['?', '!']
l3 = [str(i) + j for i, j in zip(l1, cycle(l2))]
l3
```

```
['1?', '2!', '3?', '4!', '5?']
```

```
text + '-' + 'USA'
```

```
'san francisco-USA'
```

```
import calendar
' '.join(map(str, calendar.month_name[1:6]))
```

```
'January February March April May'
```

```
','.join(map(str, [str(i) for i in range(1, 11)]))
```

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

Popular Functions

```
# count number of characters in a string
len(text)
```

```
13
```

```
# convert string to lower-case
text.lower()
```

```
'san francisco'
```

```
# convert string to upper-case
text.upper()
```

```
'SAN FRANCISCO'
```

```
# replace character in a string
text.replace('a', 'x')
```

```
'sxn frxncisco'
```

```
# substring (var[start_index(inclusive) : end_index(exclusive)])
text[2: 8]
```

```
'n fran'
```

```
# compare strings
display(text is var3)
display(text > var3)
display(text < var3)
```

```
False
```

```
True
```

```
False
```

```
# split string (default whitespace)
text.split(' ')
```

```
['san', 'francisco']
```

```
# substitute substring (all instances)
text.replace('an', 'we')
```

```
'swe frwecisco'
```

```
# substitute substring (first instance only)
text.replace('an', 'we', 1)
```

```
'swe francisco'
```

```
# abbreviate?
```

```
string = "Los Angeles, officially the City of Los Angeles and often
known   by its initials L.A.,is the second-most populous city in the
United States   (after New York City), the most populous city in
California and the county   seat of Los Angeles County. Situated in
Southern California, Los Angeles is   known for its Mediterranean
climate, ethnic diversity, sprawling metropolis,   and as a major
center of the American entertainment industry."
```

```
from textwrap import wrap

wrap(string)
```

```
['Los Angeles, officially the City of Los Angeles and often known
by',
 'its initials L.A.,is the second-most populous city in the United',
 'States   (after New York City), the most populous city in
California',
 'and the county   seat of Los Angeles County. Situated in Southern',
 'California, Los Angeles is   known for its Mediterranean climate,',
 'ethnic diversity, sprawling metropolis,   and as a major center of
the',
 'American entertainment industry.']
```

```
len(string)
```

```
438
```

```
string.lower()
```

```
'los angeles, officially the city of los angeles and often known   by
its initials l.a.,is the second-most populous city in the united
states   (after new york city), the most populous city in california
and the county   seat of los angeles county. situated in southern
california, los angeles is   known for its mediterranean climate,
ethnic diversity, sprawling metropolis,   and as a major center of the
american entertainment industry.'
```

```
string.upper()
```

```
'LOS ANGELES, OFFICIALLY THE CITY OF LOS ANGELES AND OFTEN KNOWN   BY
ITS INITIALS L.A.,IS THE SECOND-MOST POPULOUS CITY IN THE UNITED
STATES   (AFTER NEW YORK CITY), THE MOST POPULOUS CITY IN CALIFORNIA
AND THE COUNTY   SEAT OF LOS ANGELES COUNTY. SITUATED IN SOUTHERN
CALIFORNIA, LOS ANGELES IS   KNOWN FOR ITS MEDITERRANEAN CLIMATE,
ETHNIC DIVERSITY, SPRAWLING METROPOLIS,   AND AS A MAJOR CENTER OF THE
AMERICAN ENTERTAINMENT INDUSTRY.'
```

```
# replace multiple characters in a string !case sensitive
# NOTE: use ,1 in replace function for first match only
str_new = string
for c1, c2 in zip("and", "for"):
    str_new = str_new.replace(c1, c2)
str_new
```

```
'Los Aogeles, officifly the City of Los Aogeles for ofteo koowo   by
its ioitifls L.A.,is the secoor-most populous city io the Uoiter
Stftes   (ffter New York City), the most populous city io Cfliforoif
for the couoty   seft of Los Aogeles Couoty. Situfter io Southero
Cfliforoif, Los Aogeles is   koowo for its Meriterrfoefo climfte,
```

```
ethoic riversity, sprfwliog metropolis,   for fs f mfjor ceoter of the
Americfo eotertfiomeot iorustry.'
```

```
# replace whole words !case sensitive
# NOTE: use ,1 in replace function for first match only
string.replace('is', 'was')
```

```
'Los Angeles, officially the City of Los Angeles and often known   by
its initials L.A.,was the second-most populous city in the United
States   (after New York City), the most populous city in California
and the county   seat of Los Angeles County. Situated in Southern
California, Los Angeles was   known for its Mediterranean climate,
ethnic diversity, sprawling metropolwas,   and as a major center of
the American entertainment industry.'
```

```
# difference between two vectors
setA = {"monday","tuesday","wednesday"}
setB = {"monday","thursday","friday"}

setA.difference(setB)
```

```
{'tuesday', 'wednesday'}
```

```
# check if strings/sets are equal
display(setA is setB)
display(setA is setA)
```

```
False
```

```
True
```

```
# TODO: Abbreviate?
```

```
# split strings
x = ["ID-101","ID-102","ID-103","ID-104"]
[i.split('-') for i in x]
```

```
[['ID', '101'], ['ID', '102'], ['ID', '103'], ['ID', '104']]
```

Matches in Text

```
# return boundary indices of the first matched string
import re
try:
    display(re.search("Angeles", string).span())
except:
    display("no match")
```

```
(4, 11)
```

```
# Return T/F of matched string
if re.search("Angeles", string):
```

```
    display(True)
else:
    display(False)
```

```
True
```

```
# return boundary indices of all matches
for i in re.finditer("is", string):
    display(i.span())
```

```
(87, 89)
```

```
(293, 295)
```

```
(371, 373)
```

Regular Expressions

```
text = "As much mud in the streets as if the waters had but newly
retired from   the face of the earth, and it would not be wonderful to
meet a   Megalosaurus, forty feet long or so, waddling like an
elephantine lizard   up Holborn Hill."
```

```
pat = r'waters'
```

```
re.search(pat, text).span()
```

```
(37, 43)
```

```
pat = r'ing?'
```

```
display("First occurence:", re.search(pat, text).span())
```

```
display("All occurences:")
for i in re.finditer(pat, text):
    display(i.span())
```

```
'First occurence:'
```

```
(12, 14)
```

```
'All occurences:'
```

```
(12, 14)
```

```
(180, 183)
```

```
(200, 202)
```

Disjunction

```
pat = r'(waters?)|(earth)|([Hh]ill)'
```

```
for i in re.finditer(pat, text):
    display(i.span())
```

```
re.findall(pat, text)
```

```
(37, 43)
```

```
(89, 94)
```

```
(224, 228)
```

```
[('waters', '', ''), ('', 'earth', ''), ('', '', 'Hill')]
```

Metacharacters/Escape Sequences

Add in front of these characters to escape them: | ( ) [ ] { } $ * + ?

```
sample = "hello$ this is \ rahul writing reg&&x"
pat = r'\$|\\|\&'

re.findall(pat, sample)
```

```
['$', '\\', '&', '&']
```

Quantifiers

```
# .
num = "1000101011111"
# kleene plus - greedy
display(re.findall(r'10+1', num))

# kleene star - non greedy
display(re.findall(r'10*', num))

# .
display(re.findall(r'0.1', num))
display(re.findall(r'0.+1', num))
```

```
['10001', '101']
```

```
['1000', '10', '10', '1', '1', '1', '1', '1']
```

```
['001', '011']
```

```
['000101011111']
```

```
names = ' '.join(map(str,
["anna","crissy","puerto","cristian","garcia","steven","alex","rudy"])
)
display(names)

# doesn't matter if e is a match
display(re.findall(r'[^ ]*e*[^ ]*', names))
```

```
'anna crissy puerto cristian garcia steven alex rudy'
```

```
['anna',
 '',
 'crissy',
 '',
 'puerto',
 '',
 'cristian',
 '',
 'garcia',
 '',
 'steven',
 '',
 'alex',
 '',
 'rudy',
 '']
```

```
# must match t one or more times
display(re.findall(r'[^ ]*t+[^ ]*', names))
```

```
['puerto', 'cristian', 'steven']
```

```
# must match n two times
display(re.findall(r'[^ ]*n{2}[^ ]*', names))
```

```
['anna']
```

Sequences

```
string = "I have been to Paris 20 times"

# match a digit
display(re.findall(r'\d+', string))
```

```
['20']
```

```
# match a non-digit
display(re.findall(r'\D+', string))
```

```
['I have been to Paris ', ' times']
```

```
# match a space - returns positions
for i in re.finditer(r'\s+', string):
    display(i.span())
```

```
(1, 2)
```

```
(6, 7)
```

```
(11, 12)
```

```
(14, 15)
```

```
(20, 21)
```

```
(23, 24)
```

```
# match a non-space
display(re.findall(r'\S+', string))
```

```
['I', 'have', 'been', 'to', 'Paris', '20', 'times']
```

```
# match a word character
display(re.findall(r'\w+', string))
```

```
['I', 'have', 'been', 'to', 'Paris', '20', 'times']
```

```
# match a non-word character
display(re.findall(r'\W+', string))
```

```
[' ', ' ', ' ', ' ', ' ', ' ']
```

Character classes

```
string = "20 people got killed in the mob attack. 14 got severely injured"
```

```
# extract numbers
display(re.findall(r'[0-9]+', string))
```

```
['20', '14']
```

```
# extract w/o digits
display(re.findall(r'[^0-9]+', string))
```

```
[' people got killed in the mob attack. ', ' got severely injured']
```

POSIX Characters

Not natively supported by re library - need to test RegEx

```
string = "I sleep 16 hours, a day" + "I sleep 8 hours a day." + "You sleep how many hours ?"
```

```
!pip3 install regex
```

```
Requirement already satisfied: regex in /Users/rah/opt/anaconda3/lib/python3.8/site-packages (2021.4.4)
```