A string is a list of characters in order.

A character is anything you can type on the keyboard in one keystroke,

like a letter, a number, or a backslash.

Strings can have spaces: "hello world".

An empty string is a string that has 0 characters.

Python strings are immutable

Python recognize as strings everything that is delimited by quotation marks

(" " or ' ').

String Manipulation

To manipulate strings, we can use some of Pythons built-in methods.

Creation

word = "Hello World"

>>> print word

Hello World

Accessing

Use [ ] to access characters in a string

word = "Hello World"

letter=word[0]

>>> print letter

H

Length

word = "Hello World"

>>> len(word)

11

Finding

word = "Hello World"

>>> print word.count('l') # count how many times l is in the string

3

>>> print word.find("H") # find the word H in the string

0

>>> print word.index("World") # find the letters World in the string

6

Count

s = "Count, the number of spaces"

>>> print s.count(' ')

8

Slicing

Use [ # : # ] to get set of letter

Keep in mind that python, as many other languages, starts to count from 0!!

word = "Hello World"

print word[0] #get one char of the word

print word[0:1] #get one char of the word (same as above)

print word[0:3] #get the first three char

print word[:3] #get the first three char

print word[-3:] #get the last three char

print word[3:] #get all but the three first char

print word[:-3] #get all but the three last character

word = "Hello World"

word[start:end] # items start through end-1

word[start:] # items start through the rest of the list

word[:end] # items from the beginning through end-1

word[:] # a copy of the whole list

Split Strings

word = "Hello World"

>>> word.split(' ') # Split on whitespace

['Hello', 'World']

Startswith / Endswith

word = "hello world"

>>> word.startswith("H")

True

>>> word.endswith("d")

True

>>> word.endswith("w")

False

Repeat Strings

print "."\* 10 # prints ten dots

>>> print "." \* 10

..........

Replacing

word = "Hello World"

>>> word.replace("Hello", "Goodbye")

'Goodbye World'

Changing Upper and Lower Case Strings

string = "Hello World"

>>> print string.upper()

HELLO WORLD

>>> print string.lower()

hello world

>>> print string.title()

Hello World

>>> print string.capitalize()

Hello world

>>> print string.swapcase()

hELLO wORLD

Reversing

string = "Hello World"

>>> print ' '.join(reversed(string))

d l r o W o l l e H

Strip

Python strings have the strip(), lstrip(), rstrip() methods for removing

any character from both ends of a string.

If the characters to be removed are not specified then white-space will be removed

word = "Hello World"

Strip off newline characters from end of the string

>>> print word.strip('

')

Hello World

strip() #removes from both ends

lstrip() #removes leading characters (Left-strip)

rstrip() #removes trailing characters (Right-strip)

>>> word = " xyz "

>>> print word

xyz

>>> print word.strip()

xyz

>>> print word.lstrip()

xyz

>>> print word.rstrip()

xyz

Concatenation

To concatenate strings in Python use the "+" operator.

"Hello " + "World" # = "Hello World"

"Hello " + "World" + "!"# = "Hello World!"

Join

>>> print ":".join(word) # #add a : between every char

H:e:l:l:o: :W:o:r:l:d

>>> print " ".join(word) # add a whitespace between every char

H e l l o W o r l d

Testing

A string in Python can be tested for truth value.

The return type will be in Boolean value (True or False)

word = "Hello World"

word.isalnum() #check if all char are alphanumeric

word.isalpha() #check if all char in the string are alphabetic

word.isdigit() #test if string contains digits

word.istitle() #test if string contains title words

word.isupper() #test if string contains upper case

word.islower() #test if string contains lower case

word.isspace() #test if string contains spaces

word.endswith('d') #test if string endswith a d

word.startswith('H') #test if string startswith H

def is\_prime(x):

if x < 2:

return False

else:

for n in range(2, x-1):

if x % n == 0:

return False

return True

For prime

def anti\_vowel(text):

t=""

for c in text:

for i in "ieaouIEAOU":

if c==i:

c=""

else:

c=c

t=t+c

return t