Str = “guru”

Str \* 2 = “guruguru”

Replace:

oldstring = 'I like Guru99'

newstring = oldstring.replace('like', 'love')

print(newstring)

**What is Tuple in Python?**

A tuple is just like a list of a sequence of immutable python objects. The difference between list and tuple is that list are declared in square brackets and can be changed while **tuple is declared in parentheses** and cannot be changed. However, you can take portions of existing tuples to make new tuples.

**Tuple Syntax**

Tup = ('Jan','feb','march')

To write an empty tuple, you need to write as two parentheses containing nothing-

tup1 = ();

For writing tuple for a single value, you need to include a comma, even though there is a single value. Also at the end you need to write semicolon as shown below.

Tup1 = (50,);

## Packing and Unpacking

In packing, we place value into a new tuple while in unpacking we extract those values back into variables.

x = ("Guru99", 20, "Education") # tuple packing

(company, emp, profile) = x # tuple unpacking

print(company)

print(emp)

print(profile)

## Built-in functions with Tuple

To perform different task, tuple allows you to use many built-in functions like all(), any(), enumerate(), max(), min(), sorted(), len(), tuple(), etc.

## Advantages of tuple over list

* Iterating through tuple is faster than with list, since tuples are immutable.
* Tuples that consist of immutable elements can be used as key for dictionary, which is not possible with list
* If you have data that is immutable, implementing it as tuple will guarantee that it remains write-protected

Classes in Python

About using "self" in Python

* The self-argument refers to the object itself. Hence the use of the word self. So inside this method, self will refer to the specific instance of this object that's being operated on.
* Self is the name preferred by convention by Pythons to indicate the first parameter of instance methods in Python. It is part of the Python syntax to access members of objects

## Regular Expression Syntax

**RE**

import re

* "re" module included with Python primarily used for string searching and manipulation
* Also used frequently for web page "Scraping" (extract large amount of data from websites)

We will begin the expression tutorial with this simple exercise by using the expressions (w+) and (^).

## Example of w+ and ^ Expression

* **"^":**This expression matches the start of a string
* **"w+**": This expression matches the alphanumeric character in the string

import re

list = ["guru99 get", "guru99 give", "guru Selenium"]

for element in list:

z = re.match("(g\w+)\W(g\w+)", element)

if z:

print((z.groups()))

patterns = ['software testing', 'guru99']

text = 'software testing is fun?'

for pattern in patterns:

print('Looking for "%s" in "%s" ->' % (pattern, text), end=' ')

if re.search(pattern, text):

print('found a match!')

else:

print('no match')

abc = ['guru99@google.com](mailto:'guru99@google.com), [careerguru99@hotmail.com](mailto:careerguru99@hotmail.com), [users@yahoomail.com](mailto:users@yahoomail.com)'

emails = re.findall(r'[\w\.-]+@[\w\.-]+', abc)

for email in emails:

print(email)

## Strftime()

Regex:

Findall

Search