Programming is Fun

An introduction to Python

Indian Linux Users Group Coimbatore
http://ilugcbe.org.in
mail2ilugcbe@gmail.com

Our Inspiration

Kenneth Gonsalves (1953-2012)

About

Purpose

- Learn art of computer programming with one of the easiest programming language in the world.
- Get ready to enjoy the joy of programming in Python.

Warning

- This is a hands on training program.
- No theory included in the slides; it may discussed whenever and wherever it is required.
- Be patient,listen, practice and ask questions
- Do not hesitate to experiment
- Our volunteers are here to help you in practising
- Be courageous enough to try
- We are leaning Python not rocket science
- Be simple and think in a simplistic way

Are You Ready !

Why Python

Why

- Used almost everywhere
- Fun
- Concise
- Simple
- Powerful
- Filled with spices to write effective programs

Installation

Already installed in GNU/Linux systems
If you are using M$ Windows download Python from python.org
If you are downloading Python for Windows remember to download Python 2.7 only.

Interactive Interpreter

Interpreter

Contains REPL

- Read
- Evaluate
- Print
- Loop

Interpreter

```
$ python
Python 2.6.6 (r266:84292, Feb
26 2011, 23:10:42)
[GCC 4.3.4] on linux2
Type "help", "copyright",
"credits" or "license" for
more information.
>>>
```

Hello World!!

Why Hello World

Writing 'Hello World' program is the ritual to please the programming gods to be a good programmer!!

```
>>> print "Hello World"
Hello world
```

Let's Jump to Programming

Programming Practice - 1

Create a file hello.py
type print "Hello World"
Listen to the screen for instructions.
In-case of any issues just raise your hand our volunteers will be there to help you.

Programming Practice - 1

Run the program

```
$python hello.py
```

Time to make your hands dirty

Note

Make sure that you have understood the first step.
In-case of trouble we can practice it for couple of minutes.

Variables

```
age = 33                    # Integer
avg = 33.35                 # Float
name = "linux"              # String
bool = True                 # Boolean
another = "44"              # String
```

Variables - naming

- use lowercase
- use underscore between words my_age
- do not start with numbers
- do not use built-ins
- do not use keywords

Reserved

and del for is raise assert elif from lambda return break else global not try class except if or while continue exec import pass yield def finally in print

Time to make your hands dirty

Everything is an object

- Everything in python is an object

An object has

- identity (id)
- type (type)
- value (mutable or immutable)

```
>>> age = 33
>>> type(age)
<type 'int'>
>>> id(age)
167263248
>>> age 34
>>> id(age)
167263236
```

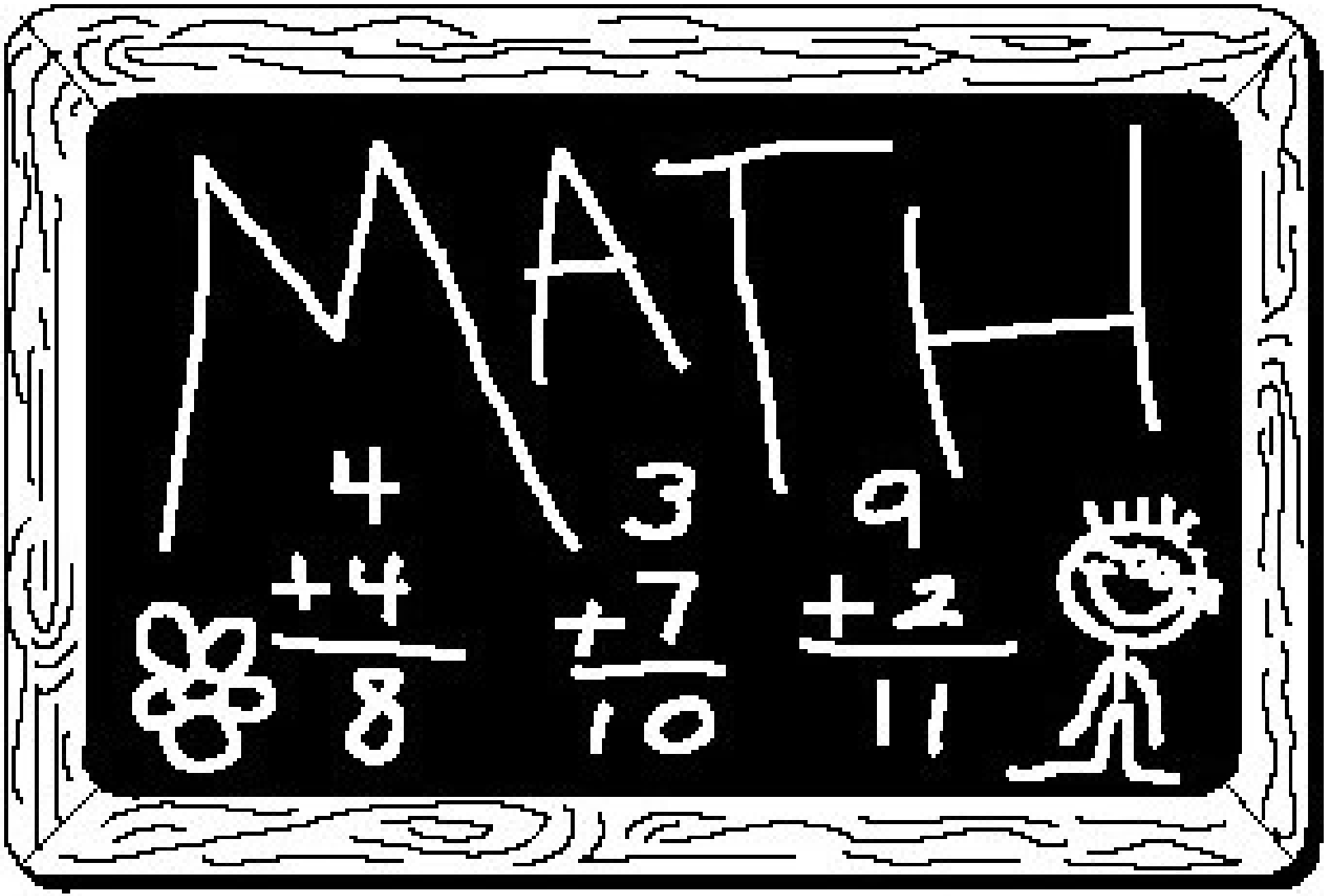
Casting

```
>>> age = 33
>>> str(age)
'33'
>>> float(age)
33.0
>>> long(age)
33L
>>> int(age)
33
```

Time for some maths

+ , - , * , ** (power),% (modulo), // (floor division), < (less than) > greater than, <=, >=, ==

It is maths time now



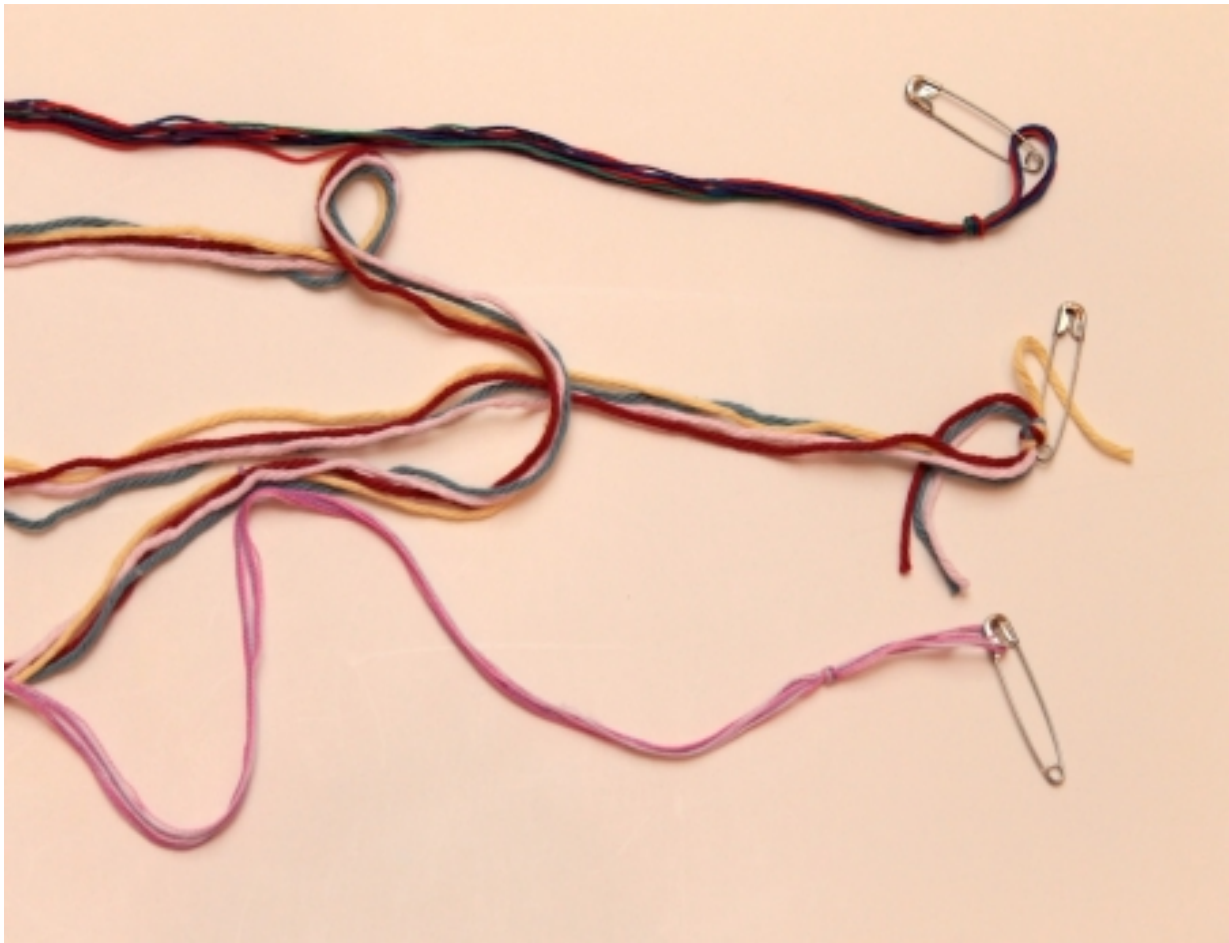
String

```
>>> name = 'linux'
>>> address = "Coimbatore 1st street"
>>> description = """ We are teaching python to
young chaps"""
>>> with_new = "this sentence have one \n new line"
```

String

```
>>> name = "linux"
>>> nameu = name.upper()
>>> namel = nameu.lower()
>>> namel.find('l')
>>> ",".join(name)
>>> name.startswith('l')
>>> name.endswith('x')
>>> namet = name.title()
>>> wspace = "     with space "
>>> stripped = wspace.strip()
```

Playing with String



Comments

- Single line comments starts with #
- Multi-line comments should be with in """ """

```
>>> avg = 45.34 #float
>>> name = "ilug-cbe" """ This is a
multiline comment """
```

Boolean

```
>>> t = True
>>> f = Flase

>>> n = None
```

Conditionals

equality,greater,less,is, is not ...

$==$, $!=$, $>$, $>=$, $<$, $<=$, is, is not

Conditionals

```
>>> 1 == 1
>>> 1 >= 0
>>> 0 <= 1
>>> 0 != 1
>>> "Rajani" is "rajani"
>>> "Vijay" is not "Rajani"
>>> name = None
>>> if name is not None:
...     #do something
```

Boolean operators

Used to combine conditional logic

- and
- or
- not

if ...

```
if mark > 60 and mark < 100:
  print "First Class"
elif mark < 60:
  print "Second Class Only :-("
else:
  print "Ooops !!!!"
```

if ... elif time

Sequences

list

List is an ordered collection of objects.

```
names = ["rms","linus","guido","larry"]
marks = [45,50,60,80,90]
mixed = ["linux",12,12.35,90L]
```

Sequences - list

```
names = []
print names
names.append("linus")
print names
numbers = [6,9,2,3,1,8,4]
print len(numbers)
print numbers.sort()
print numbers.reverse()
another = [9,3,6]
numbers.extend(another)
print numbers
numbers.insert(0,20)
print numbers
```

Sequences - list

```
print numbers[0]
print numbers[-1]
print numbers[2:-2]
print numbers[:-2]
print numbers[2:]
print numbers[:]
```

Sequences -tuple

tuple

Tuple is like list only. But tuple is immutable

```
nums = (1,2,3,4)
print nums
print len(nums)
print nums[0]
print nums[-1]
```

Sequences range

```
nums = range(20)
print nums
selected = range(20,60)
print selected
jump2 = range(10,100,2)
print jump2
```

Let's do some sequencing

Iteration

```
names = ["linus","rms","guido","larry"]
for name in names:
  print "hello %s" %(name)

for num in range(10,20,2):
  print num
```

Iteration

```
for i in range(len(names)):
  print i,names[i]

for i,v in enumerate(names):
  print i,v
```

Iteration break,continue and pass

```
for name in names:
  print name
  if name == "guido":
    break
```

```
for name in names:
  print name
  if name == "linus":
    continue
```

```
for name in names:
  print name
  if name == "rms":
    pass
```

Iteration - while

```
my_number = 10

while my_number < 20:
    print my_number
    my_number += 2
```

Let's Iterate

Dictionaries

Also called as hash, hashmap or associative array

```
address = {"name":"ILUG-CBE","houseno":"Nil",
"street":"any whare","city":"Coimbatore"}
print address
```

Dictionaries

```
address["state"] = "tamilnadu"
address["web"] = "ilugcbe.org.in"

print address
```

Dictionaries

```
print address.has_key("country")

print address.get("phone","No Phone Number Provided")
```

Dictionaries

```
address.setdefault("country","India")

print address

print address.keys()

print address.values()

print address.items()

del address["country"]

print address
```

Let's practice Dictionaries

Functions

```
def say_hai():
  print "hello"

say_hai()
```

Functions

```
def add_two(num):
  return num + 2

res = add_two(4)

print res
```

Functions

```
def add(a,b):
    """
    adds two numbers
    """
    return a + b

res = add_two(4,5)

print res
```

Functions

```
def demo(*args):
  """
  *args demo
  """
  for arg in args:
    print i * 2

demo(1,2,3,4,5,6)
```

Functions

```
def demo(num,*args):
  """
  *args demo
  """
  for arg in args:
    print i * num

demo(1,2,3,4,5,6)
```

Functions

```
def demo(num,*args):
  """
  *args demo
  """
  mul = []
  for arg in args:
    mul.append(i * num)

  return sum(mul)

res = demo(2,2,3,4,5,6)

print res
```

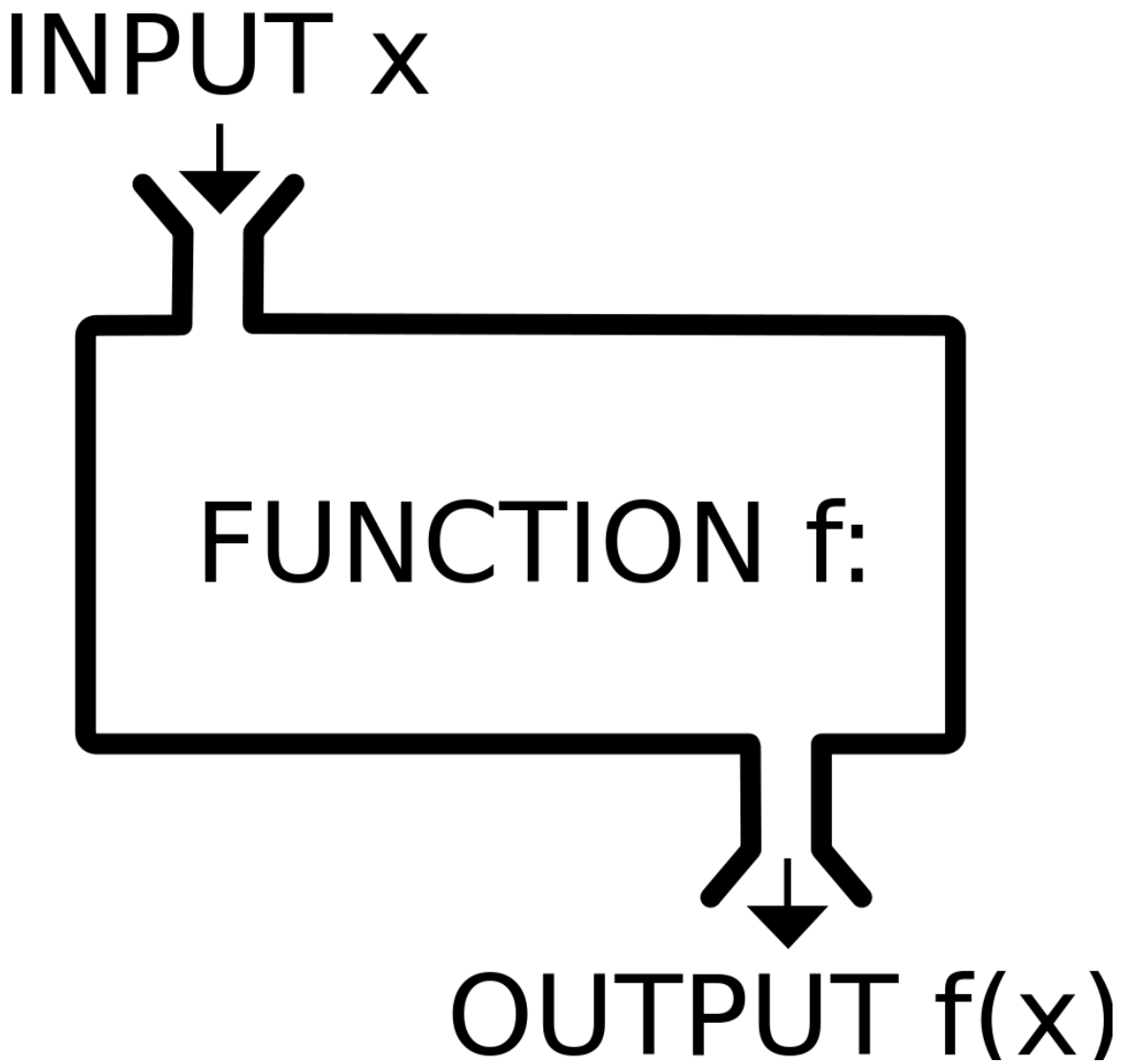
Functions

```
def marker(roll_no,details):
  if details['marks'] > 60:
    print "Roll no %d %s" %(roll_no, "First Class")

marker(12,marks = 62)
```

Functions

```
def mulbythree(num,three=3):
    """
    default argument example
    """
    return num * three

res = mulbythree(43)
print res
```

Be functional now



Tricks 1

```
nums = [1,2,3,4,5,6,7,8,9]
mulbt = [num * 2 for num in nums]

print nums
print mulbt
```

Tricks - 2

```
nums = [1,2,3,4,5,6,7,8,9]
mulbt = [num * 2 for num in nums if num / 2 != 0]

print nums
print mulbt
```

Tricks - 3

```
nums = [1,2,3,4,5,6,7,8,9]
numtuple = tuple(nums)

print type(nums)
print type(numtuple)
```

Tricks - 4

```
print "ILUGCBE".lower().upper().title()
```

User input

```
name = raw_input("Tell your name: ")

print name

age = int(raw_input("Tell your age: "))

print age
```

Tricks time

Lambda

```
product = lambda x,y : x * y

res = product(12,13)
print res
```

Lambda

```
dbtnt = lambda x : x * 2 if x % 2 == 0 else x

res = dbtnt(12)
print res
```

Lambda

```
bors = lambda x: x > 100 and 'big' or 'small'

for i in (1, 10, 99, 100, 101, 110):
    print i, 'is', f(i)
```

Object Oriented Programming - basics

```
class MyClass:
    """
    This is my class
    """
    def __init__(self):
        #nothing

    def say_hai(self):
        print "Hai"

obj = MyClass()
obj.say_hai()
```

Object Oriented Programming - basics

```
class MyClass(object):
    """
    This is my class
    """
    def __init__(self):
        #nothing

    def say_hai(self):
        print "Hai"

obj = MyClass()
obj.say_hai()
```

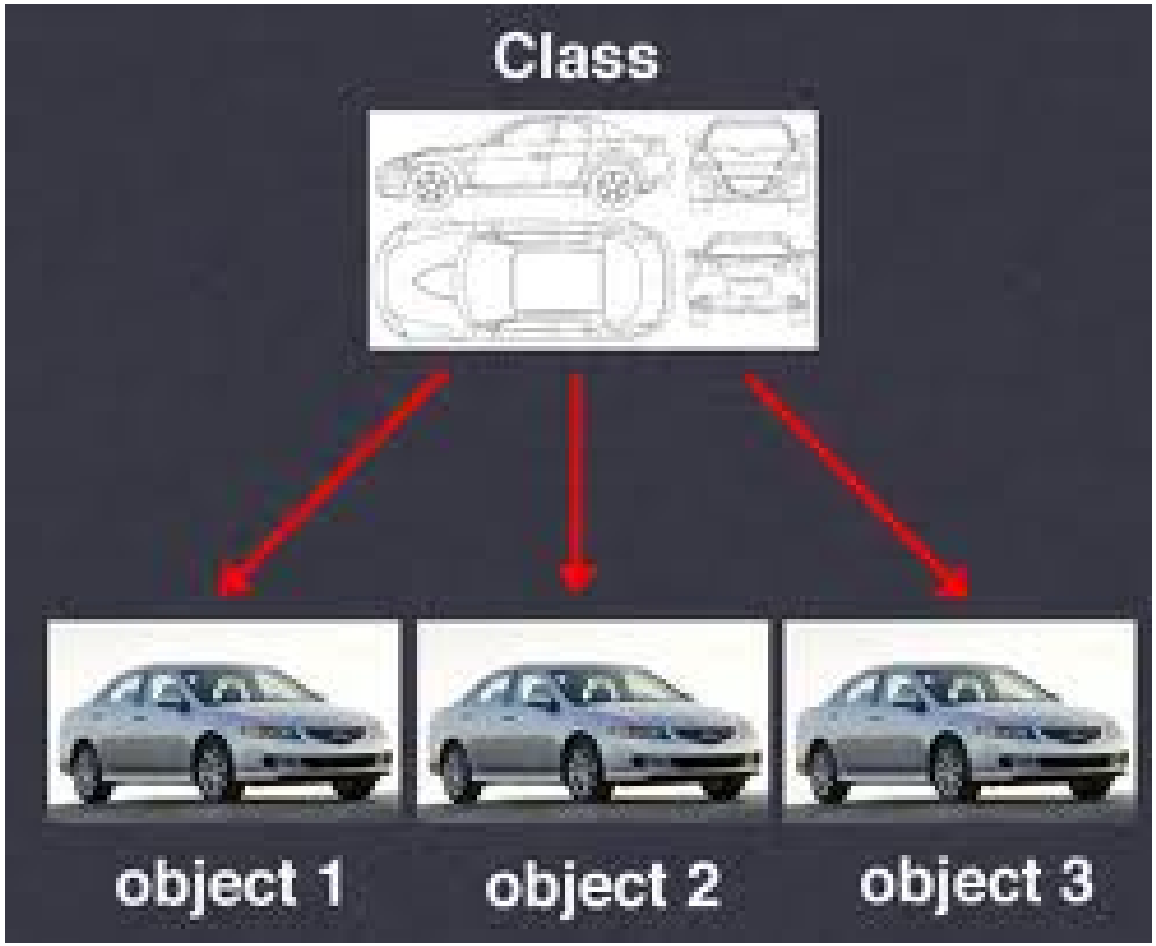
Object Oriented Programming - basics

```
class Student:
    """
    Student class
    """

    def __init__(self):
        self.college = "My College"

    def greet(self,name):
        """
        Function to greet student
        """
        print "Hello %s from %s" %(name, self.college)

student = Student()
student.greet("Jaganadh")
```

Object Oriented Programming - inheritance

```
class BeStudent:
    def __init__(self):
        Student.__init__(self)
        self.class = "BE First Year"

    def greet(self,name):
        print "Hello %s from %s %s class" %(name,
        self.college,self.class)

 student = BeStudent()
 student.greet("Jaganadh G")
```

Object Oriented Time



File I/O - read file

```
input = open("file.txt",'r')
contents = input.read()
input.close()

print contents

input = open("file.txt",'r')
contents = input.readlines()
input.close()

print contents
```

File I/O - write to file

```
names = ["linus","rms","larry","guido"]
output = open("out_file.txt",'w')
for name in names:
    output.write(name)
    output.write("\n")
```

File I/O - write to file

```
names = ["linus","rms","larry","guido"]
output = open("out_file.txt",'a')
for name in names:
    output.write(name)
    output.write("\n")
```

Let's play with files

Batteries

Standard Libraries

Python comes with batteries included. Lots of useful libraries are there in the language. Let's see some of these libraries now

```
import math
print math.pi
print math.sqrt(10)
print math.factorial(10)
print math.sin(10)
print math.pow(10,2)
print math.log(10)
print math.log(10,2)
print math.log10(10)
print math.exp(10)
```

Batteries - sys

```
import sys
print sys.platform

afl = sys.argv[1]

print sys.version
print sys.maxint
print sys.maxsize
```

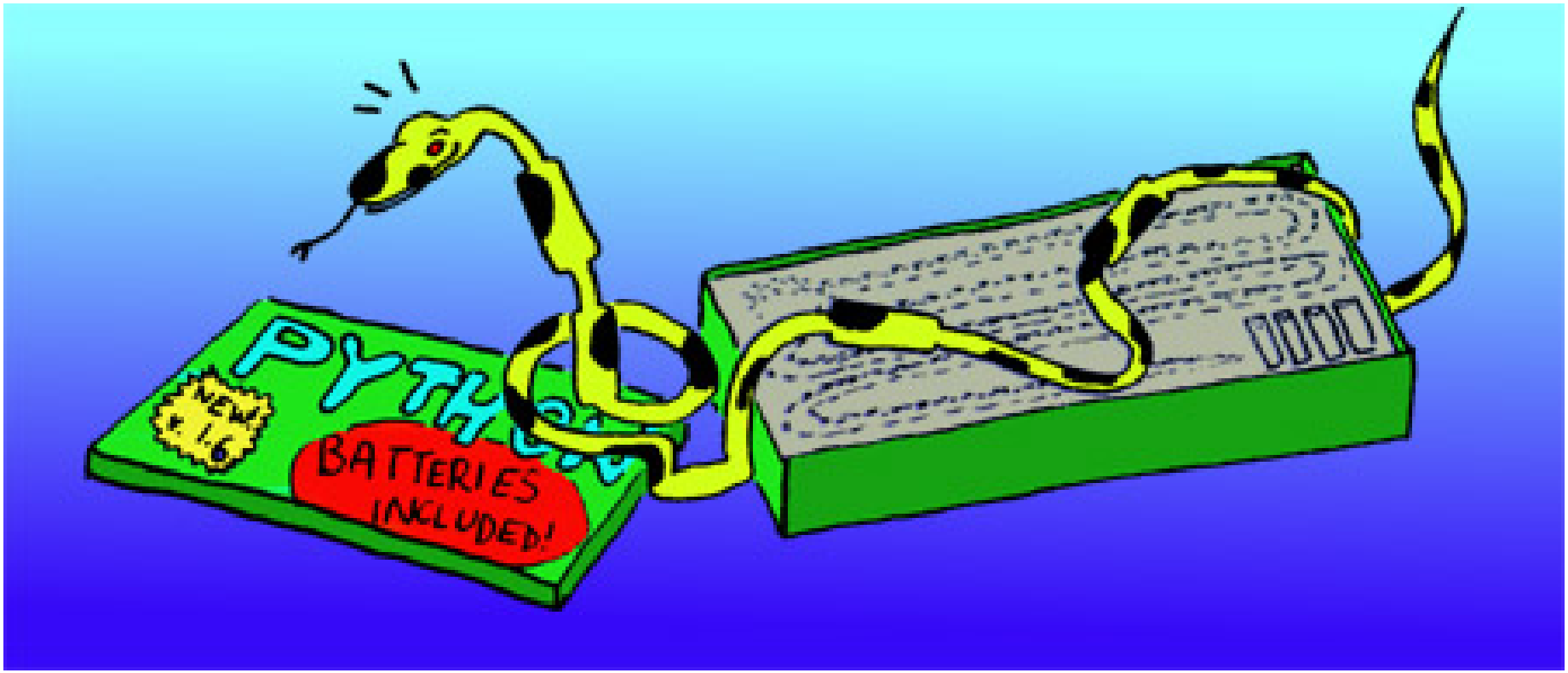
Batteries - os

```
import os
print os.curdir()
print os.getlogin()
print os.getcwd()
print os.name
```

Batteries - time,random

```
import time
print time.ctime()
print time.gmtime()

import random
print random.random()
print random.choice([1,2,3,4,5,6])
```

Battery Recharge



Question Time

Contributors

- Jaganadh G @jaganadhg
- Biju B @bijubk
- Satheesh Kumar D
- Sreejith S @tweet2sree
- Rajith Ravi @chelakkandupoda

Contact US

Web

http://ilugcbe.org.in
mail mail2ilugcbe@gmail.com