



Python – An Introduction

Shrinivasan T

tshrinivasan@gmail.com

<http://goinggnu.wordpress.com>



Python is a Programming Language

There are **so many**
Programming
Languages.

Why Python?



```
#include <stdio.h>
int main(void)
{
    printf("Hello, World!\n");
    return 0;
}
```

C/C++

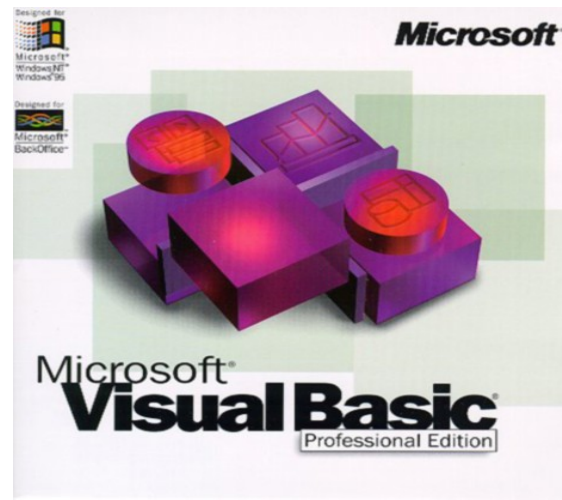


lisp



Ruby

C#



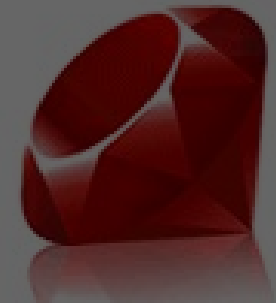
Microsoft
.net

```
#include <stdio.h>
int main(void)
{
    printf("Hello, World!\n");
    return 0;
}
```

C/C++



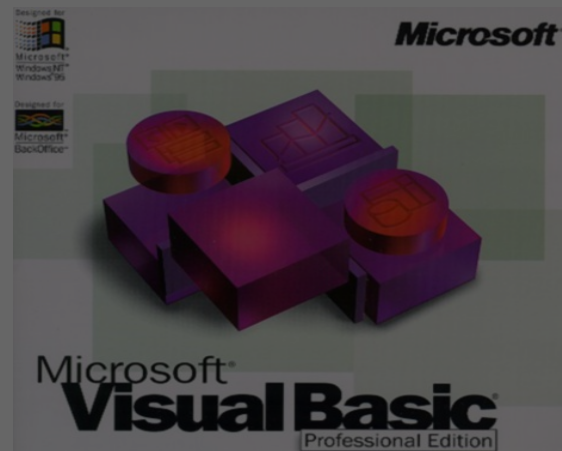
lisp



Ruby



C#



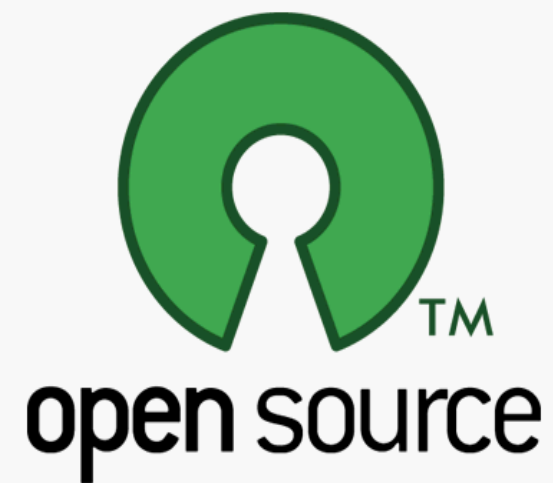
Microsoft®
.net™

The background of the slide features two vibrant orange cosmos flowers in the lower-left quadrant, set against a clear, bright blue sky. The flowers have multiple layers of petals and prominent yellow centers with dark stamens. A semi-transparent purple rounded rectangle is positioned in the lower-right area, containing the text.

Python is simple and beautiful



Python is Easy to Learn



 floss.

Python is Free Open Source Software

Can Do

- Text Handling
- System Administration
- GUI programming
- Web Applications
- Mobile Apps
- Database Apps
- Scientific Applications
- Games
- Data Analysis
- NLP
- ML
- AI
- LLM
- ...



H*istory*



Guido van Rossum
Father of Python
1991

Perl	Java	Python	Ruby	PHP
1987		1991	1993	1995



What is Python?

Python is...

A dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write.

Quick and Easy

- Interpreted Scripting Language
- Variable declarations are unnecessary
- Variables are not typed
- Syntax is simple and consistent
- Memory management is automatic

Object Oriented Programming

Classes

Methods

Inheritance

Modules

etc.,

```
def add5(x):  
    return x+5
```

```
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]),ast[0])  
    print ' %s [label="%s' % (nodename, label),  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '- %s' % ast[1]  
        else:  
            print ''  
    else:  
        print ''  
        children = []  
        for n, child in enumerate(ast[1:]):  
            children.append(dotwrite(child))  
        print ' %s -> {' % nodename,  
        for name in children:  
            print '%s' % name,
```

Examples!



```
print ("Hello World")
```

No Semicolons !

Variables

colored_index_cards

No Need to Declare Variable Types !

Python Knows Everything !

```
value = 10
```

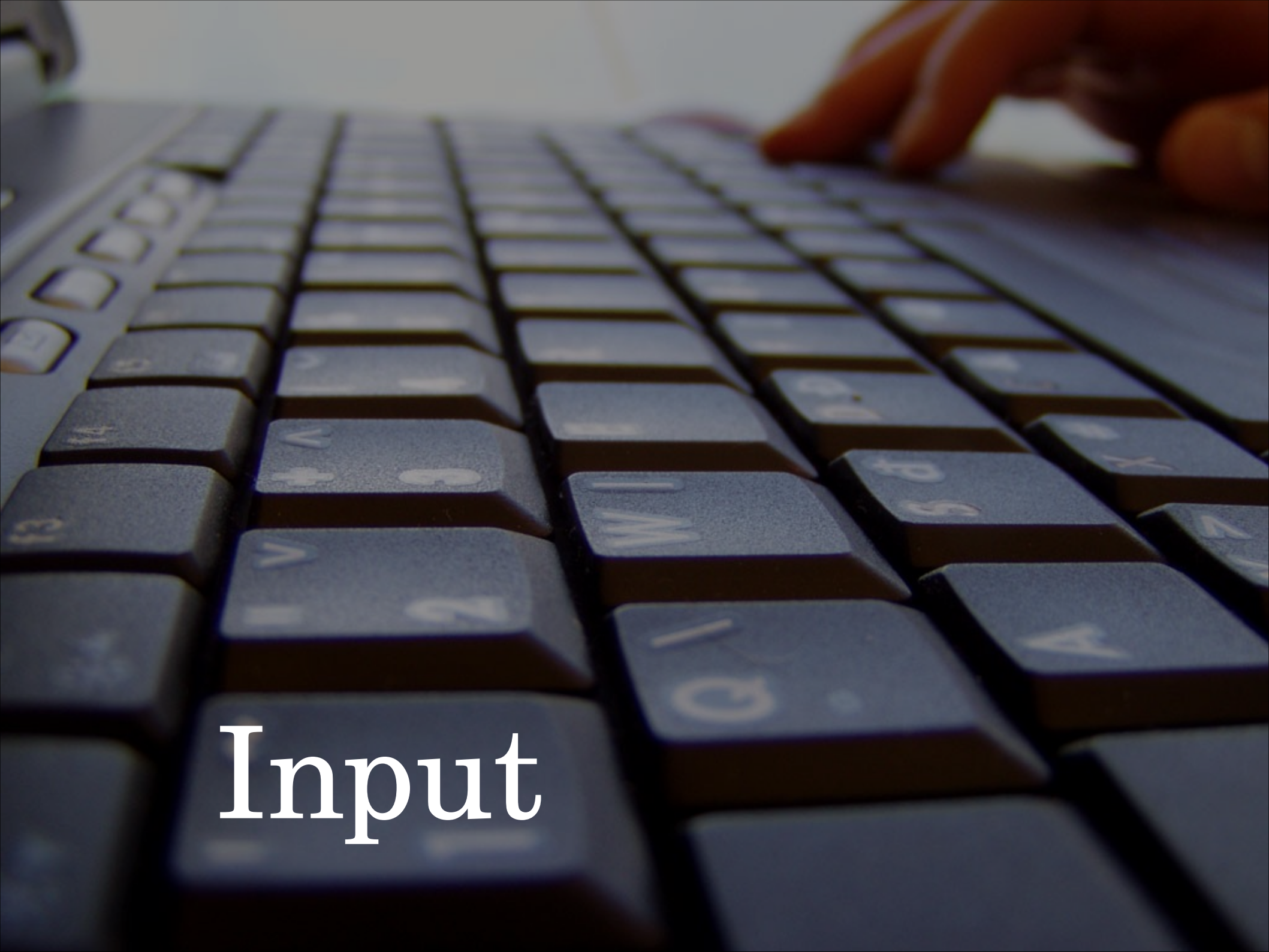
```
print(value)
```

```
value = 100.50
```

```
print(value)
```

```
value = "This is String"
```

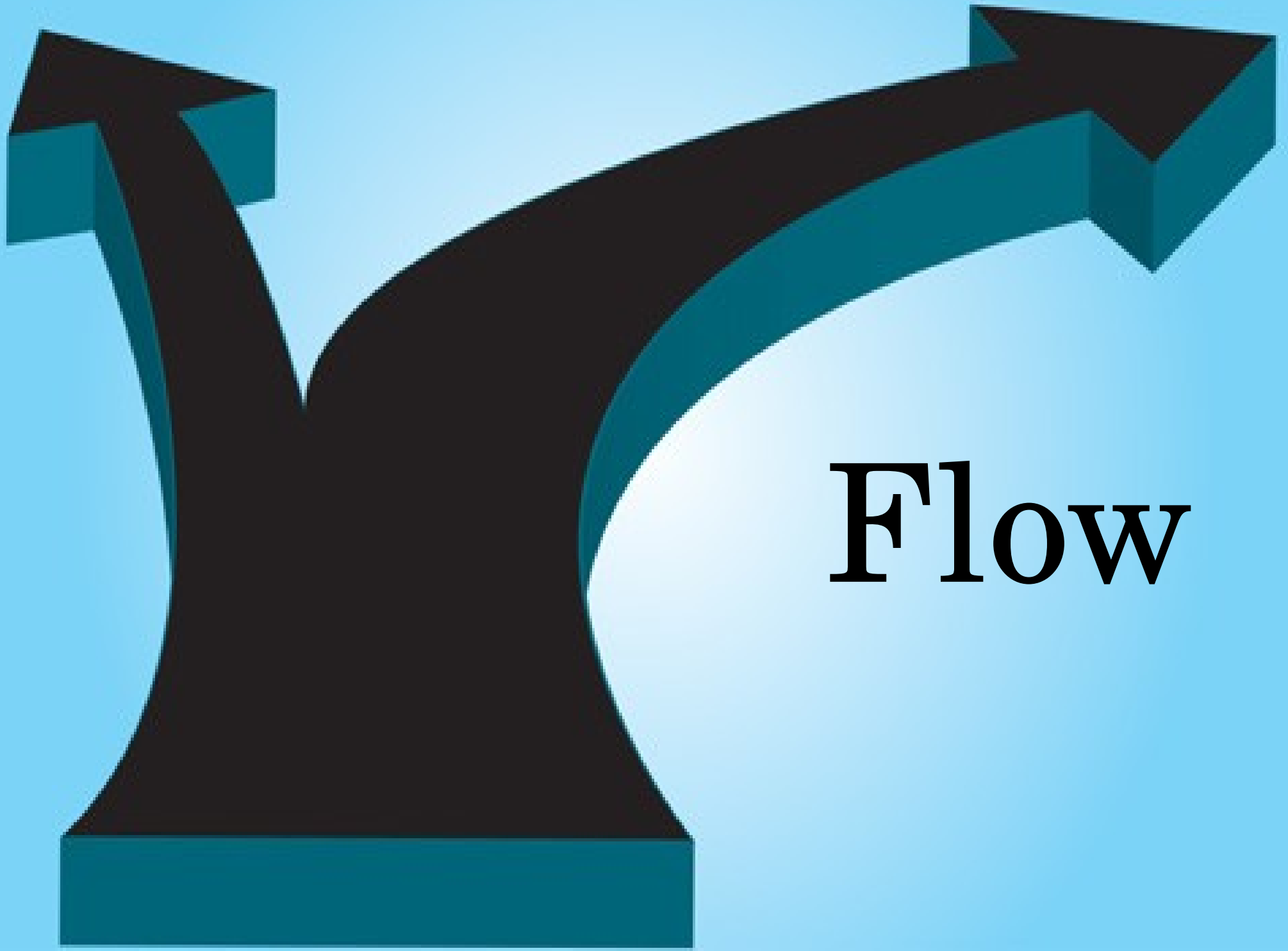
```
print(value * 3)
```



Input

```
name = input("what is Your name?")
```

```
print("Hello" , name , "Welcome")
```



Flow

```
score=int(input(" type a number"))
if score >= 5000 :
    print("You win!")
elif score <= 0 :
    print("Game over.")
else:
    print("Current score:",score)

print("Done\n")
```

Indentation



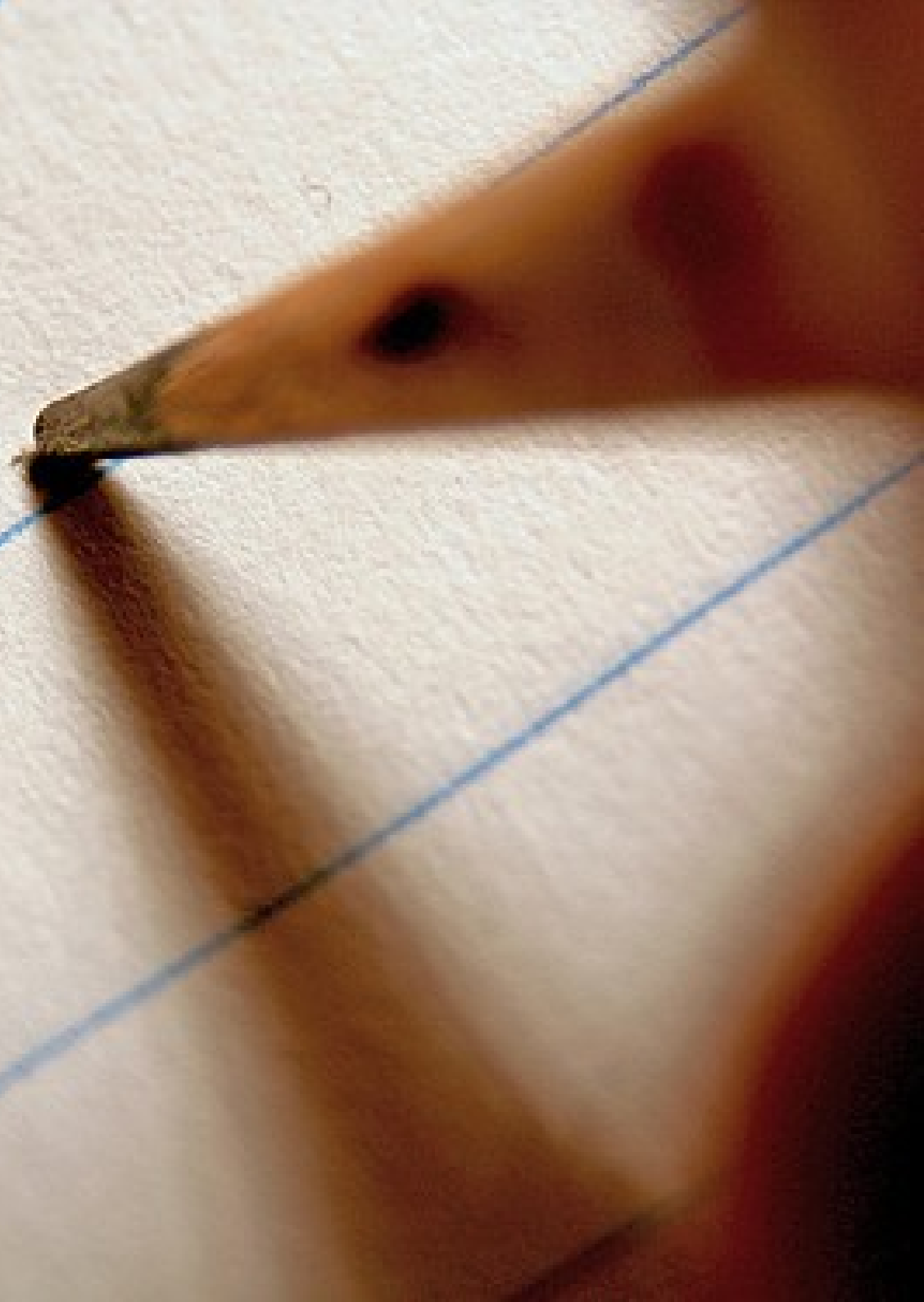
You have to follow
the Indentation
Correctly.

Otherwise,

Python will beat
you !



long





Discipline

gives

best



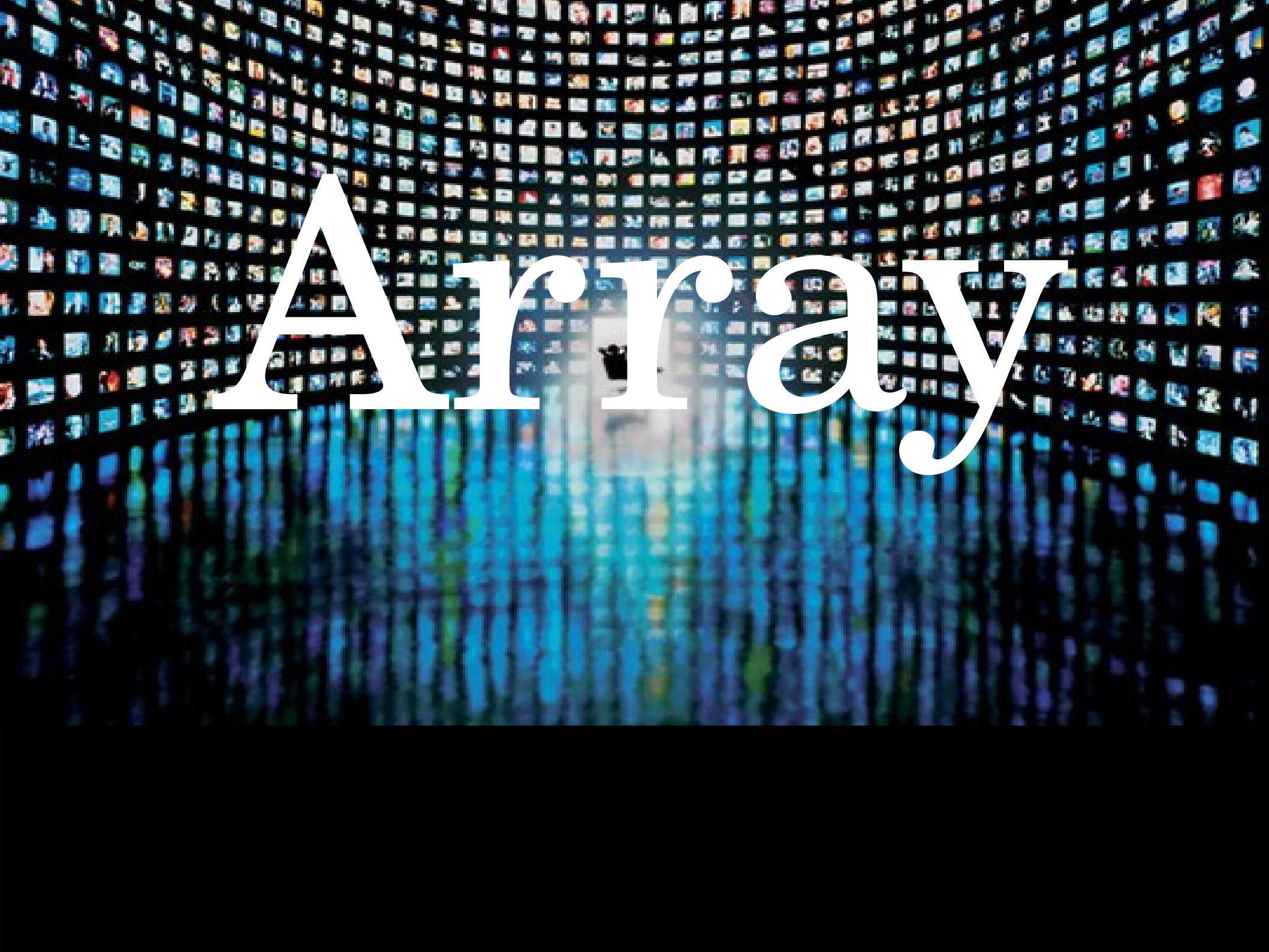
Loop

```
for i in range(1, 5):  
    print (i)  
  
print('The for loop is over')
```

```
number = 23
running = True
while running :
    guess = int(input('Enter an integer : '))
    if guess == number :
        print('Congratulations, you guessed it.')
        running = False
    elif guess < number :
        print('No, it is a little higher than that.')
    else:
        print('No, it is a little lower than that.')

print('Done')
```

Array



List = Array

```
numbers = [ "zero", "one", "two", "three",  
"FOUR" ]
```

List = Array

```
numbers = [ "zero", "one", "two", "three",  
"FOUR" ]
```

```
numbers[0]
```

```
>>> zero
```

```
numbers[4]
```

```
>>> FOUR
```

```
numbers[-1]
```

```
>>> FOUR
```

```
numbers[-2]
```

```
>>> three
```

Multi Dimension List

```
numbers = [ ["zero", "one"], ["two", "three",  
"FOUR" ]]
```

```
numbers[0]
```

```
>>> ["zero", "one"]
```

```
numbers[0][0]
```

```
>>> zero
```

```
numbers[-1][-1]
```

```
>>> FOUR
```

```
len(numbers)
```

```
>>> 2
```

Sort List

```
primes = [ 11, 5, 7, 2, 13, 3 ]
```

Sort List

```
primes = [ 11, 5, 7, 2, 13, 3 ]
```

```
primes.sort()
```

Sort List

```
primes = [ 11, 5, 7, 2, 13, 3 ]
```

```
>>> primes.sort()
```

```
>>> print(primes)  
[2, 3, 5, 7, 11, 13]
```

Sort List

```
names = [ "Shrini", "Bala", "Suresh",  
"Arul"]
```

```
names.sort()  
print(names)  
>>> ["Arul", "Bala", "Shrini", "Suresh"]
```

```
names.reverse()  
print(names)  
>>> ["Suresh", "Shrini", "Bala", "Arul"]
```

Mixed List

```
names = [ "Shrini", 10, "Arul", 75.54]
```

```
names[1]+10
```

```
>>> 20
```

```
names[2].upper()
```

```
>>> ARUL
```

Mixed List

```
names = [ "Shrini", 10, "Arul", 75.54]
```

```
names[1]+10
```

```
>>> 20
```

```
names[2].upper()
```

```
>>> ARUL
```

Append on List

```
numbers = [ 1,3,5,7]
```

```
numbers.append(9)
```

```
>>> [1,3,5,7,9]
```

Tuples

immutable

```
names = ('Arul','Dhastha','Raj')
```

```
name.append('Selva')
```

Error : Can not modify the tuple

Tuple is immutable type

String

```
name = 'Kaniyam'
```

```
name[0]
```

```
>>>'K'
```

```
myname = 'Kaniyam' + 'Foundation'
```

```
>>> 'KaniyamFoundation'
```

split

```
name = 'This is python string'
```

```
name.split(' ')
```

```
>>> ['This', 'is', 'python', 'string']
```

```
comma = 'Shrini,Arul,Suresh'
```

```
comma.split(',')
```

```
>>> ['Shrini', 'Arul', 'Suresh']
```

join

```
li = ['a','b','c','d']
```

```
s = '-'
```

```
new = s.join(li)
```

```
>>> a-b-c-d
```

```
new.split('-')
```

```
>>> ['a','b','c','d']
```

```
'small'.upper()
```

```
>>>'SMALL'
```

```
'BIG'.lower()
```

```
>>>'big'
```

```
'mIxEd'.swapcase()
```

```
>>>'MiXwD'
```

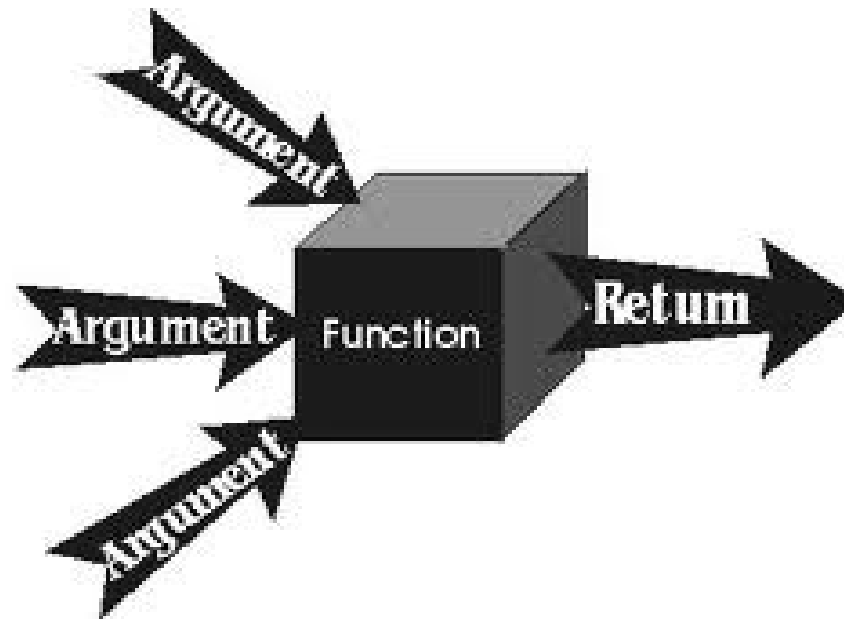


Dictionary

```
menu = {  
    "idly"           : 2.50,  
    "dosai"          : 10.00,  
    "coffee"         : 5.00,  
    "ice_cream"      : 5.00,  
    100              : "Hundred"  
}
```

```
menu["idly"]  
2.50
```

```
menu[100]  
Hundred
```



Function

```
def sayHello():  
    print('Hello World!') # block belonging of fn  
# End of function
```

```
sayHello() # call the function
```

```
def printMax(a, b):  
    if a > b:  
        print(a, 'is maximum')  
    else:  
        print(b, 'is maximum')  
printMax(3, 4)
```



File Handling

File Reading



```
f= open('poem.txt','r')  
for line in f.readlines():  
    print(line)  
f.close()
```



File Writing

```
poem = """ Programming is fun
When the work is done
if you wanna make your work also fun:
    use Python!
"""
```

```
f = open('poem.txt', 'w') # open for 'w'riting
f.write(poem) # write text to file
f.close()
```

Using in built Modules

```
#!/usr/bin/python
# Filename: using_sys.py
import time

print('The sleep started')
time.sleep(3)
print('The sleep finished')
```

```
#!/usr/bin/python
```

```
import os
```

```
print(os.listdir('/home/shrini'))
```

```
print(os.walk('/home/shrini'))
```

Making Our Own Modules

```
#!/usr/bin/python
# Filename: mymodule.py
def sayhi():
    print("Hi, this is mymodule speaking.")
version = '0.1'

# End of mymodule.py
```

```
#!/usr/bin/python
```

```
# Filename: mymodule_demo.py
```

```
import mymodule
```

```
mymodule.sayhi()
```

```
print('Version', mymodule.version)
```

```
#!/usr/bin/python
# Filename: mymodule_demo2.py
from mymodule import sayhi, version
# Alternative:
# from mymodule import *

sayhi()
print('Version', version)
```

Class

Classes

```
class Person:  
    pass # An empty block
```

```
p = Person()
```

```
print(p)
```

Classes

```
class Person:  
    def sayHi(self):  
        print('Hello, how are you?')
```

```
p = Person()
```

```
p.sayHi()
```

Classes

```
class Person:
    def __init__(self, name):
        #like constructor
        self.name = name
    def sayHi(self):
        print('Hello, my name is', self.name)

p = Person('Arulalan.T')

p.sayHi()
```

Classes

Inheritance

Classes

```
class A:
    def hello(self):
        print (' I am super class ')
class B(A):
    def bye(self):
        print(' I am sub class ')

p = B()
p.hello()
p.bye()
```

Classes

```
class A:
```

```
    Var = 10
```

```
    def __init__(self):
```

```
        self.public = 100
```

```
        self._protected_ = 'protected'
```

```
        self.__private__ = 'private'
```

```
Class B(A):
```

```
    pass
```

```
p = B()
```

```
p.__protected__
```

Database Intergration



```
import psycopg2
```

```
conn = psycopg2.connect(" dbname='pg_database'  
user='dbuser' host='localhost' password='dbpass' ")
```

```
cur = conn.cursor()  
cur.execute("""SELECT * from pg_table""")  
rows = cur.fetchall()  
print(rows)
```

```
cur.close()  
conn.close()
```

```
import psycopg2
```

```
conn = psycopg2.connect(" dbname='pg_database'  
user='dbuser' host='localhost' password='dbpass' ")
```

```
cur = conn.cursor()
```

```
cur.execute("insert into pg_table values(1,'python')")
```

```
conn.commit()
```

```
cur.close()
```

```
conn.close()
```

THE END

of code :-)

How to learn ?

Python – Shell

Interactive Python

Instance Responce

Learn as you type

bpython
ipython } teach you very easily

Editors

- Emacs
- Vim
- notepad++
- spyder
- pycharm
- jupyter
- geany
- vscode
- gedit
- kate
- Google Colab
- online-python.com

Python can communicate
With
Other
Languages

C

+

Python



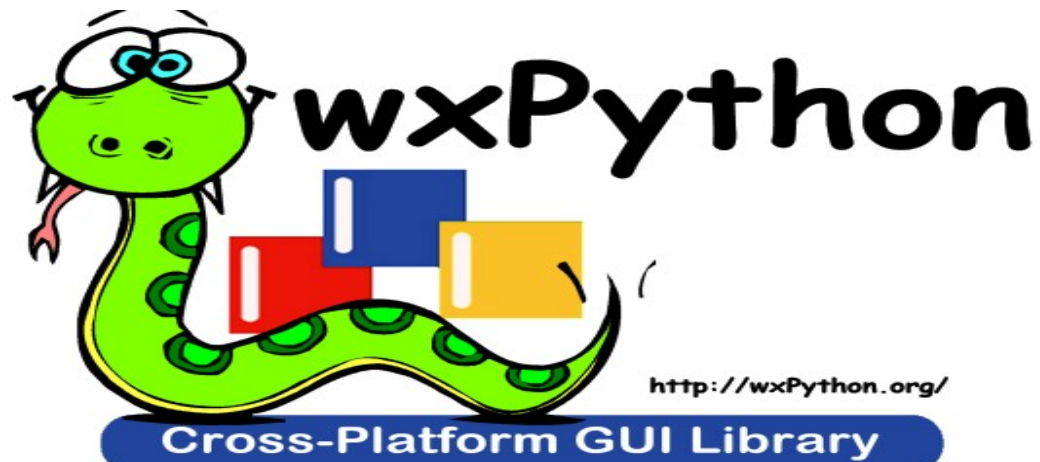
Java
+
Python



GUI

With

Python



Glade

+

Python

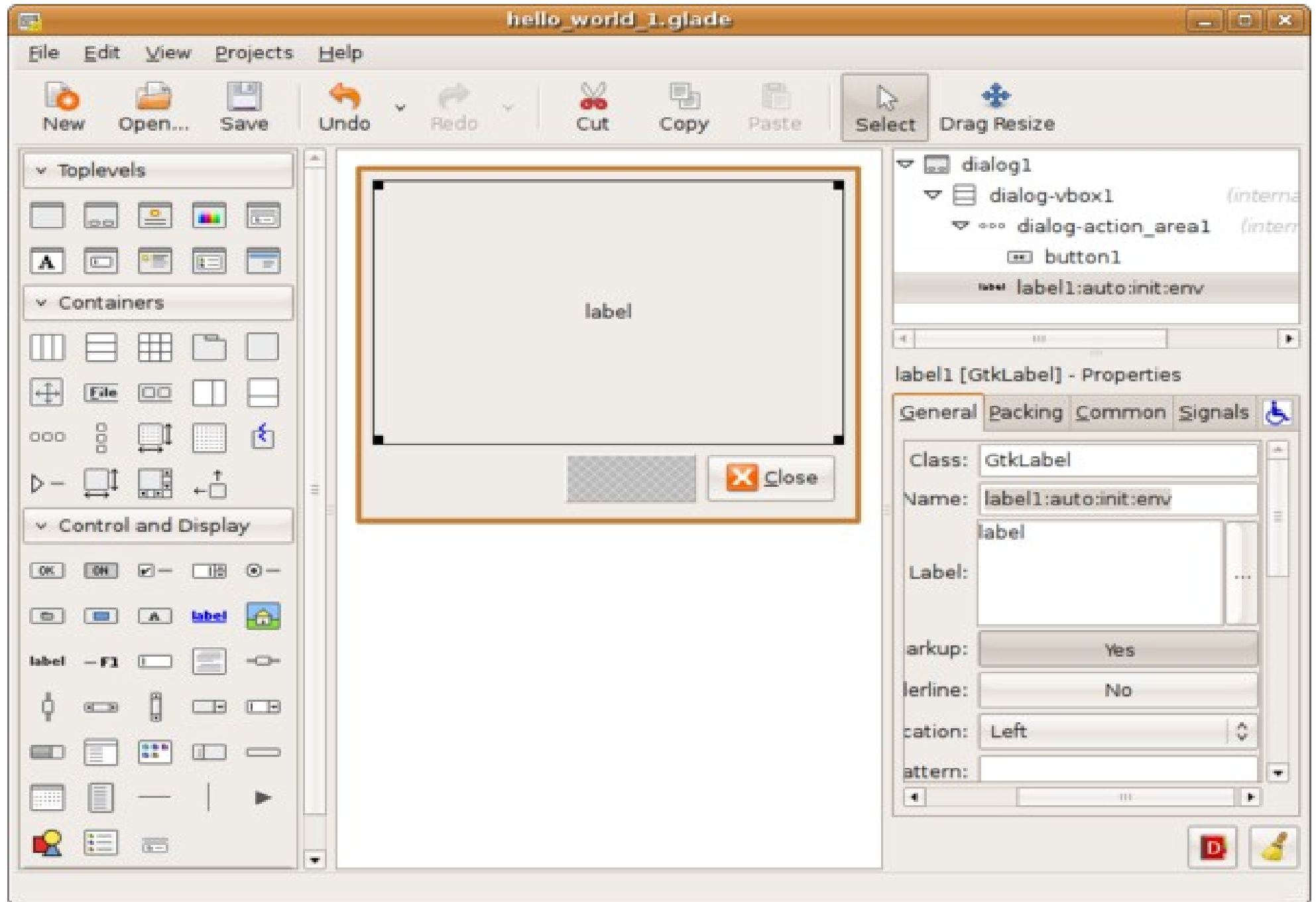
+

GTK

=

GUI APP

GLADE





Using Glade + Python



Web

django

Web Frame Work in Python



search

[Advanced Search](#)

Screen styles
normal* large userpref

- ABOUT >>
- NEWS >>
- DOCUMENTATION >>
- DOWNLOAD >>
- COMMUNITY >>
- FOUNDATION >>
- CORE DEVELOPMENT >>

- Help
- Package Index
- Quick Links (2.7)
- >> Documentation
 - >> Windows Installer
 - >> Source Distribution
- Quick Links (3.1.2)
- >> Documentation
 - >> Windows Installer
 - >> Source Distribution
- Python Jobs
- Python Merchandise

Python Programming Language – Official Website

Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.

Python is free to use, even for commercial products, because of its OSI-approved [open source license](#).

New to Python or choosing between Python 2 and Python 3? Read [Python 2 or Python 3](#).

The [Python Software Foundation](#) holds the intellectual property rights behind Python, underwrites the [PyCon conference](#), and funds other projects in the Python community.

[Read more, -or- download Python now](#)

» **Python 2.6.6rc1 released**
A [release candidate for Python 2.6.6](#) has been released for testing. Python 2.6.6 final is schedule for release on August 16, 2010.

Published: Wed, 04 Aug 2010, 18:09 -0400

» **Python 3.2 alpha 1 released**
The [first alpha release of Python 3.2](#) has been released for testing.

Published: Sun, 01 Aug 2010, 8:30 +0200

» [Frank Willison Award goes to Christian Tismer](#)

U. of St Andrews uses Python...

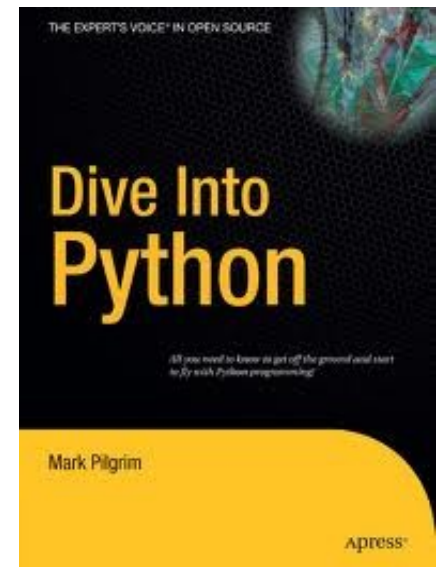
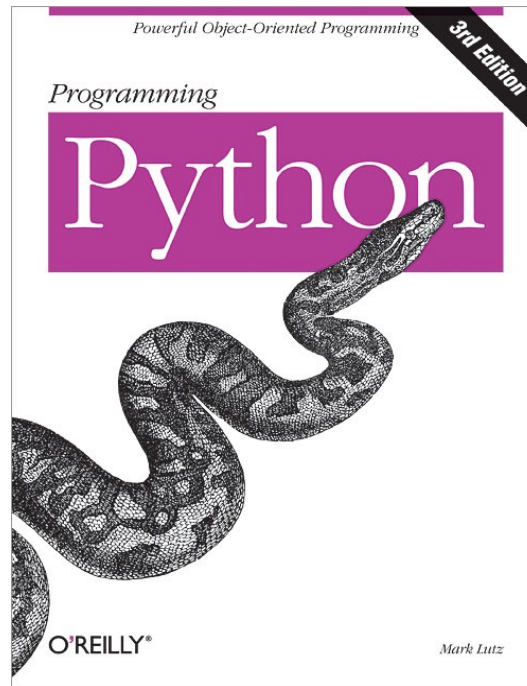
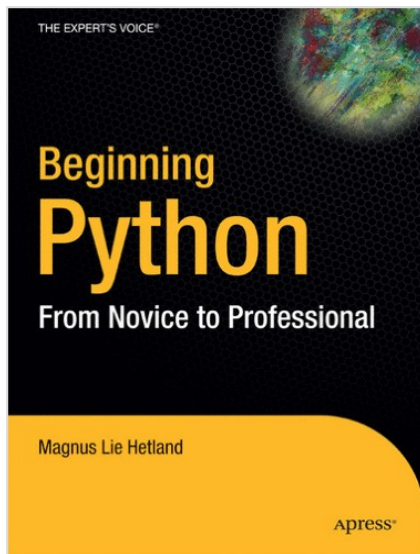


... joining users such as [Rackspace](#), [Industrial Light and Magic](#), [AstraZeneca](#), [Honeywell](#), and many others.

What they are saying...

EVE Online:

"Python enabled us to create [EVE Online](#), a massive multiplayer game, in record time. The EVE Online server cluster runs over 25,000 simultaneous players in a shared space simulation, most of which is



A Byte of Python

Swaroop C H

<https://python.swaroopch.com/>

Easiest free online book for
Python

Type to search

- Introduction
- Dedication
- Preface
- About Python
- Installation
- First Steps
- Basics
- Operators and Expressions
- Control flow
- Functions
- Modules
- Data Structures
- Problem Solving
- Object Oriented Programming
- Input and Output
- Exceptions
- Standard Library
- More
- What Next
- Appendix: FLOSS
- Appendix: About
- Appendix: Revision History

A Byte of Python

"A Byte of Python" is a free book on programming using the Python language. It serves as a tutorial or guide to the Python language for a beginner audience. If all you know about computers is how to save text files, then this is the book for you.

For Python version 3

This book will teach you to use Python version 3. There will also be guidance for you to adapt to the older and more common Python version 2 in the book.

Who reads A Byte of Python?

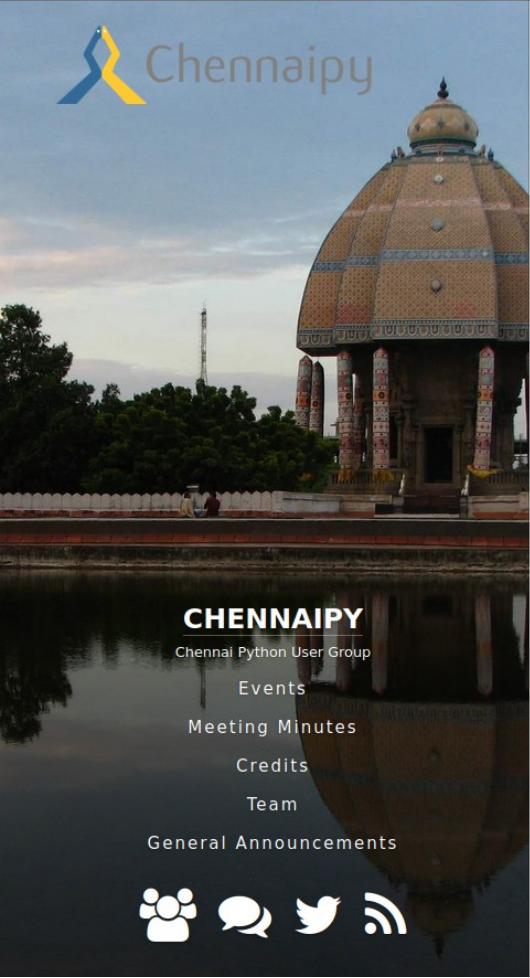
Here are what people are saying about the book:

Somewhere around 2004 - 05 when I was convinced I wasn't smart enough to be a programmer , I came came across the original A Byte of #Python, and that changed my entire perspective on computing and life , I owe a lot to that book @swaroopch had written. -- [Rahul](#) on Jul 30, 2020

This is the book that got me into programming almost a decade ago. Thank you @swaroopch. You changed my life. -- [Stefan Froelich](#) on Aug 2, 2019

I am writing this email to thank you for the great help your book has done for me! It was a really good book that I enjoyed thoroughly. As a 15 year old who has never done programming before, trying to learn Python online was difficult and I couldn't understand anything. But I felt like your book gave was much easier to understand and eased me into the whole new world of programming. Thanks to you, I can now write a high level language with ease. I thought programming would be hard and boring, but with your book's help, I realised how fun and interesting yet simple it can be! I would like to thank you again for your hard work on helping out beginners like me. -- [Prattvashita Tahivat](#) on Sep 17 2019





WELCOME!

Chennaipy is a Python user-group, where Pythonistas in and around Chennai come together to share their knowledge and to contribute to the promotion of Python.

Upcoming Events: [PyCon India 2019, Chennai](#)



Mailing List
[Join](#) | [Archives](#)



Meetup
[Join](#) | [Members](#)



Github
[Projects](#) | [Teams](#)



Twitter
[Follow](#) | [Tweets](#)



Slack
[Join](#) | [Discuss](#)



Chennai PyCon Mailing List
[Join](#) | [Archives](#)

Join ChennaiPy Mailing List and Ask/Answer questions

<https://mail.python.org/mailman/listinfo/chennaipy>



