

PTN-101 Python programming

Section	Contents
Introduction to Python	An introduction to the Python programming language. Covers details of how to start and stop the interpreter and write programs. Introduces Python's basic datatypes, files, functions, and error handling.
Working with Data	A detailed tour of how to represent and work with data in Python. Covers tuples, lists, dictionaries, and sets. Students will also learn how to effectively use Python's very powerful list processing primitives such as list comprehensions. Finally, this section covers critical aspects of Python's underlying object model including variables, reference counting, copying, and type checking.
Program Structure, Control Flow, Functions, and Exceptions	More information about how to organize larger programs and how to reuse code. Covers more advanced aspects of Python control-flow and expands upon how to define and use functions. Also covers scoping rules, documentation strings, anonymous functions, exec, and eval.
Modules, Packages, and the Standard Library	How to organize programs into modules and packages. Expands upon further details of Python variable scoping rules and use of namespaces. This section concludes with an overview of some of the most commonly used library modules and instructions on how to install third party library modules.
Classes	An introduction to object-oriented programming in Python. Describes how to create new objects, overload operators, and utilize Python special methods.
Inside the Python Object Model	A detailed look at how objects are implemented in Python. Major topics include the definition of objects, the object lifecycle, object representation, attribute binding, inheritance, and special properties of classes including descriptors, properties, slots, private attributes, static methods, and class methods.
Working with Objects	A more detailed look at how to effectively utilize objects in Python. Topics include encapsulation of data and algorithms, how to organize inheritance hierarchies, abstract classes, and useful design patterns. A major theme of this section focuses on how to implement code so that there is a loose-coupling between different software components. In addition, we will explore object oriented programming techniques used throughout the Python standard library.
Object Persistence	How to save, restore, copy, and serialize objects. Covers the marshal, pickle, shelve, and struct modules. The programming API used to interface with relational databases such as MySQL, Oracle, and Sybase is also discussed.
Documentation, Testing, and Debugging	This section discusses many issues that are considered important to Python software development. This includes effective use of documentation strings and program testing using both the doctest and unittest modules. The Python debugger and profiler are also described. Finally, a number of tips are given for improving program reliability.
Iterators, Generators, Closures, and Decorators	Covers the iteration protocol, iterable objects, generators, generator expressions, and concepts related to "lazy" evaluation of Python statements. A major focus of this section concerns the use of iteration for solving various problems in systems programming (e.g., processing large datafiles and infinite data streams).
Working with Text	A more in-depth look at how to efficiently parse and generate text data in Python. Topics include string splitting, string stripping, string joining, and template strings. This section also covers the re module, Python's module for regular expression pattern matching and substitution.

PTN-101 Python programming

Files and the file system	A more detailed look at file I/O in Python. Topics include how to work with files and directories in a portable manner as well as unicode I/O. Also covers useful modules for working with a wide variety of common file formats such as zip, gzip, bzip, tar, and CSV.
Working with Processes	This section covers details of the Python interpreter process including command line options, environment variables, and I/O. The problem of interacting with other processes and applications from Python is then discussed with detailed coverage of the subprocess module.
Concurrency	Detailed coverage on different options for writing concurrent programs in Python. Topics include threads, message queues, co-routines, and co-processes. Special emphasis will be given to a technique of writing concurrent programs that is highly portable and easy to adapt to a variety of different system configurations.
Configuration and logging	This section focuses on two important problems that are often overlooked in systems programming classes--namely how to deal with application configuration files and how to add a logging capability to a program. Special attention is given to the ConfigParser and logging modules.
Network Fundamentals	An introduction to the basic concepts of network programming. Covers the essential details of TCP/IP and the essential concepts of socket programming.
Server side programming	Modules related to the implementation of Internet servers. Covers the SocketServer module. In addition, you will learn how to quickly write HTTP and XML-RPC servers.
Client-side programming	Detailed coverage of modules that allow you to interact with a wide variety of standard Internet services. These include HTTP, FTP, SMTP, and XML-RPC. Special attention will be given to the urllib2 module that allows Python to interact with web servers.
Internet Data Handling	An overview of modules related to processing common Internet data formats. Covers a number of low-level data encodings such as base64 and quopri. Special attention is given to parsing data associated with the web and email.
XML	A survey of XML processing options in Python. Discuss the three major XML interfaces; SAX2, DOM, and the ElementTree API.