



PROFESSIONAL LINUX CONSULTING & TRAINING

CURSO AVANZADO DE PYTHON

OBJETIVO

Que el estudiante conozca qué es Python, aprenda a diseñar, crear, optimizar y realizar aplicaciones con las herramientas gráficas y de consola.

TEMARIO

Chapter 16: **Scopes and Arguments**

- Scope Rules
- The global Statement
- Scopes and Nested Functions
- Passing Arguments
- Special Argument-Matching Modes
- Chapter Summary

Chapter 17: **Advanced Function Topics**

- Anonymous Functions: lambda
- Applying Functions to Arguments
- Mapping Functions over Sequences: map
- Functional Programming Tools: filter and reduce
- List Comprehensions Revisited: Mappings
- Iterators Revisited: Generators
- Timing Iteration Alternatives
- Function Design Concepts
- Function Gotchas
- Chapter Summary
- BRAIN BUILDER

Chapter 18: **Modules: The Big Picture**

- Why Use Modules?
- Python Program Architecture
- How Imports Work
- Chapter Summary

Chapter 19: **Module Coding Basics**

- Module Creation
- Module Usage
- Module Namespaces
- Reloading Modules
- Chapter Summary

Chapter 20: **Module Packages**

- Package Import Basics
- Package Import Example
- Why Use Package Imports?
- Chapter Summary

Chapter 21: **Advanced Module Topics**

- Data Hiding in Modules
- Enabling Future Language Features
- Mixed Usage Modes: `__name__` and `__main__`
- Changing the Module Search Path
- The import as Extension
- Relative Import Syntax
- Module Design Concepts
- Module Gotchas
- Chapter Summary
- BRAIN BUILDER

Chapter 22: **OOP: The Big Picture**

- Why Use Classes?
- OOP from 30,000 Feet
- Chapter Summary

Chapter 23: **Class Coding Basics**

- Classes Generate Multiple Instance Objects
- Classes Are Customized by Inheritance
- Classes Can Intercept Python Operators
- The World's Simplest Python Class
- Chapter Summary

Chapter 24: **Class Coding Details**

- The class Statement
- Methods
- Inheritance
- Operator Overloading
- Namespaces: The Whole Story
- A More Realistic Example
- Chapter Summary

Chapter 25: **Designing with Classes**

- Python and OOP
- Classes As Records
- OOP and Inheritance: "Is-a" Relationships
- OOP and Composition: "Has-a" Relationships
- OOP and Delegation
- Multiple Inheritance
- Classes Are Objects: Generic Object Factories
- Methods Are Objects: Bound or Unbound
- Documentation Strings Revisited
- Classes Versus Modules
- Chapter Summary

Chapter 26: **Advanced Class Topics**

- Extending Built-in Types
- Pseudoprivate Class Attributes
- New-Style Classes
- Static and Class Methods
- Function Decorators
- Class Gotchas
- Chapter Summary
- BRAIN BUILDER

Chapter 27: **Exception Basics**

- Why Use Exceptions?
- Exception Handling: The Short Story
- The try/except/else Statement
- The try/finally Statement
- Unified try/except/finally
- The raise Statement
- The assert Statement
- with/as Context Managers
- Chapter Summary

Chapter 28: **Exception Objects**

- String-Based Exceptions
- Class-Based Exceptions
- General raise Statement Forms
- Chapter Summary

Chapter 29: **Designing with Exceptions**

- Nesting Exception Handlers
- Exception Idioms
- Exception Design Tips
- Exception Gotchas
- Core Language Summary
- Chapter Summary

- BRAIN BUILDER

Appendix : **Installation and Configuration**

- Installing the Python Interpreter
- Configuring Python

Appendix : **Solutions to End-of-Part Exercises**

- Part I, Getting Started
- Part II, Types and Operations
- Part III, Statements and Syntax
- Part IV, Functions
- Part V, Modules
- Part VI, Classes and OOP
- Part VII, Exceptions and Tools

<p>Duración: 25 horas Material: DVD de Fedora Core Linux y CD de programas y utilerías de Python Documentación: Manual Oficial de Python</p>
--