

Day 1:

1. 1. Introduction to Jython
 1. Jython Requirements
 2. Welcome to Jython
 1. Starting Jython
 2. Adding Items to the Window
 3. Adding Behavior
 3. What's Jython Good For?
 4. The Benefits of Jython
 1. For Java Programmers
 2. For Python Programmers
 - Quiz
2. 2. Jython Basics
 1. Running Jython
 2. Basic Types
 3. Operators
 4. Numeric Types
 1. Integer
 2. Long
 3. Float
 4. Complex
 5. Numerical Functions
 5. Sequences: Lists and Tuples
 1. Sequence Access
 2. Sequence Functions and Functions
 3. Functional Programming
 4. Tuples: Immutable Sequences
 6. Strings
 1. String Formatting
 2. String Functions
 7. Mappings and Dictionaries
 8. Jython Files
3. 3. Jython Control Flow
 1. Statements and Expressions
 2. Assignment
 1. Unpacking Assignment
 2. Augmented Assignment
 3. Printing
 4. Blocks
 5. Conditional Logic
 6. Loops
 7. List Comprehensions
 8. Exceptions
 9. Evaluating Code Dynamically
4. 4. Modules and Functions
 1. Modules
 2. Functions
 3. Parameter Passing with Style

1. Scoping Rules
 2. Flying First Class
 4. Import Statements and Packages
 1. Import Statements
 2. Importing Java Classes
 1. Auto-loading through lookup
 3. Reload
5. 5. Object-Oriented Jython
 1. Creating Classes
 2. Methods
 3. Classes, Instances, and Access Control
 1. Class and Static Attributes
 2. Access Control
 3. Get and Set Methods
 4. Using `__getattr__`
 5. Using `__setattr__`
 4. Inheritance
 5. Special Methods
 1. General Instance Special Methods
 2. Sequences and Mappings
 3. Numbers
6. 6. Using Java from Jython
 1. Basic Object Usage
 2. Automatic Type Conversion
 1. Managing Overloaded Methods
 2. Basic Data Conversions
 3. Converting Arrays
 3. Java Arrays and the `jarray` Module
 4. Java-to-Python Types: Automatic Conversion
 5. Collection Wrappers
 6. Avoiding Name Collisions
7. 7. Inheriting from Java
 1. Basic Mechanics
 2. Java Interoperation and Overriding
 3. Calling Super Methods and Constructors
8. 8. Reflection and JavaBeans
 1. Bean-Based Reflection
 1. Using Beans in Jython
 2. Indexed Properties
 3. Event Properties
 2. Python-Based Reflection
 1. Reflecting on Java Objects
 2. Active Reflection
9. 9. Using Swing
 1. Java Versus Jython
 2. A Calculator
 3. An HTML Browser
 4. An HTML Source Browser
10. 10. Using Java Libraries
 1. JDBC and the Python Database API
 1. Using `zxJDBC`

- 2. Connection Objects
 - 3. Cursor Objects
- 2. Servlets and PyServlet
 - 1. Using PyServlet
 - 2. A Python Template Tool
 - 3. Reloading Servlet Modules
- 3. Jython and XML
 - 1. SAX and DOM
 - 2. JDOM and Jython
- 11. 11. Jython Standard Library
 - 1. Using Python Modules
 - 2. System and File Modules
 - 1. The sys Module
 - 2. The os Module
 - 3. The os.path module
 - 4. File Pattern Matching with glob
 - 3. Regular Expressions
 - 1. The re Module
 - 2. Match Objects
 - 3. Regular Expression Objects
 - 4. Special Characters
 - 4. Serialization and Pickling
 - 5. Unit Testing with PyUnit
 - 1. Simple Tests
 - 2. Tests in Groups
- 12. 12. Embedding Jython Inside Java
 - 1. Setting Up an Interpreter
 - 2. Executing Code
 - 3. Accessing the Interpreter Namespace
 - 4. Using PyObjects
 - 5. PyObject Subclasses
 - 6. Catching Exceptions
 - 7. Embedding Examples
- 13. 13. Compiling Jython
 - 1. Why Compile?
 - 2. The Compiler in Action
 - 1. [Java Source](#)
 - 2. Top-Level Java
 - 3. Packaging in Java Archives
 - 4. Using Jython in Java
 - 5. A Simple Applet
- 14. A. Installing Jython
- 15. B. Jython Options and Registry
 - 1. Jython Options
 - 2. Jython Compiler Options
 - 3. The Jython Registry
- 16. C. Jython Exceptions
- 17. D. Jython and CPython Differences
 - 1. Jython Extensions
 - 2. Major Design Differences
 - 3. Minor Design Differences

4. Differences Due to Java Details
5. Differences Due to Java Limitations
6. Minor Issues
7. Java Objects Versus Python Objects