The Python Tutorial

Lina Liu

Oct, 19, 2022

Why Python

1. Powerful
2. Efficient high-level data structures
3. Simple but effective OOP
4. Ideal scripting language
5. Free third party python modules, tools, programs
6. Easily extended in other languages

What in python Tutorial

1. Basic concepts and noteworthy features
2. Lots of examples
3. Extensions:
    1. Standard objects and modules: python standard library
    2. Language definition: python language reference
    3. Write extensions in C/C++: Extending and Embedding the python interpreter/ Python API reference manual

Other References

1. Description of standard objects and modules
    1. The Python Standard Library
2. Formal definition of the language
    1. The Python Language Reference
3. Extensions in C and C++
    1. Extending and Embedding the Python Interpreter
4. The Glossary is also worth going through
    1. >>> the default python prompt of the interactive shell
    2. Code examples can be executed interactively in the interpreter
5. Books covering python in depth
    1. Effiective python
    2. Python cookbook
    3. Fluent Python

What you will get?

1. Give you a good idea of the languages's flavor and style
2. You will be able to read and write python modules and programs
3. Be ready to the next step
    1. Learn more about the various python library modules described in The python standard library

Contents

Shining Python

1. Automate you task.
    1. Perform a search-and-replace over a large number of text files
    2. Rename and rearrange a bunch of photo files
    3. Write a small custom database
    4. A specialized GUI application
    5. Simple game
2. Compare with C/C++/java, the burdens of C/C++/Java
    1. You may find the usual write/compile/test/re-compile cycle of above languages is too slow
    2. Writing the test code a tedious task
    3. When you use an extension language, we may design/implement a whole new language
3. Compare with Unix Shell, the burdens of Unix Shell
    1. Shell scripts are best at moving around files and changing text data, but
    2. Not well-suited for GUI applications or games.
4. Compare with Awk or Perl
    1. Python offer much more structure and support for large programs.
    2. Python offers much more error checking than C
    3. Python has high-level data types built in.
5. Easily integrated modules
    1. Sprit your programs into modules that can be reused in other python programs
6. Interpreted language
    1. No compilation and linking is necessary.
    2. Interpreter can be used interactively.
7. Tidy and readably language
    1. Express complex operations in a single statement
    2. No beginning and ending brackets
    3. No variable or argument declaration are necessary
8. Python is extensible
    1. Once you are hooked, you can link the python interpreter into an application written in C
9. How the python name come from
    1. Named after the BBC show. “Monty Python’s Flying Circus”