PLCT S.A de C.V

CURSO PYTHON

Descripción General

Objetivo

Que el estudiante aprenda a desarrollar scripts y programas con Python mediante el uso y aplicación n de los tipos de datos, funciones, módulos, librerias y metodologías orientadas a objetos que proporciona el lenguaje.

Duración

50 Horas

Requisitos

www.cursoslinux.com.mx
ventas@plct.com.mx
PLCT S.A. de C.V.
Tel.: 55 4522 7839/551800 7696/7224447684

PLCT S.A de C.V

TEMARIO

1. About Python
    1.1. Why should I use Python?
    1.2. What Python does well
    1.3. What Python doesn't do as well
1. Getting started
    1.1. ◦ Installing Python
    1.2. ◦ IDLE and the basic interactive mode
    1.3. ◦ Using IDLE's Python Shell window
    1.4. ◦ Hello, world
    1.5. ◦ Using the interactive prompt to explore Python
1. 1 . The Quick Python overview
    1.1. ◦ Python synopsis
    1.2. ◦ Built-in data types
    1.3. ◦ Control flow structures
    1.4. ◦ Module creation
    1.5. ◦ Object-oriented programming
1. The essentials
    1.1. ◦ The absolute basics
    1.2. ◦ Indentation and block structuring
    1.3. ◦ Differentiating comments
    1.4. ◦ Variables and assignments
    1.5. ◦ Expressions
    1.6. ◦ Strings
    1.7. ◦ Numbers
    1.8. ◦ The None value
    1.9. ◦ Getting input from the user
    1.10. ◦ Built-in operators
    1.11. ◦ Basic Python style
1. Lists, tuples, and sets
    1.1. ◦ Lists are like arrays
    1.2. ◦ List indices
    1.3. ◦ Modifying lists
    1.4. ◦ Sorting lists
    1.5. ◦ Other common list operations
    1.6. ◦ Nested lists and deep copies
    1.7. ◦ Tuples
    1.8. ◦ Sets

www.cursoslinux.com.mx
ventas@plct.com.mx
PLCT S.A. de C.V.
Tel.: 55 4522 7839/551800 7696/7224447684

PLCT S.A de C.V

1. Strings
    1.1. ◦ Strings as sequences of characters
    1.2. ◦ Basic string operations
    1.3. ◦ Special characters and escape sequences
    1.4. ◦ Converting from objects to strings
    1.5. ◦ Using the format method
    1.6. ◦ Formatting strings with %
    1.7. ◦ Bytes
1. Dictionaries
    1.1. ◦ What is a dictionary?
    1.2. ◦ Other dictionary operations
    1.3. ◦ Word counting ◦
    1.4. What can be used as a key?
    1.5. ◦ Sparse matrices
    1.6. ◦ Dictionaries as caches
    1.7. ◦ Efficiency of dictionaries
1. Control flow
    1.1. ◦ The while loop
    1.2. ◦ The if-elif-else statement
    1.3. ◦ The for loop
    1.4. ◦ List and dictionary comprehensions
    1.5. ◦ Statements, blocks, and indentation
    1.6. ◦ Boolean values and expressions
    1.7. ◦ Writing a simple program to analyze a text file
1. Functions
    1.1. ◦ Basic function definitions
    1.2. ◦ Function parameter options
    1.3. ◦ Mutable objects as arguments
    1.4. ◦ Local, nonlocal, and global variables
    1.5. ◦ Assigning functions to variables
    1.6. ◦ lambda expressions
    1.7. ◦ Generator functions
    1.8. ◦ Decorators
1. . Modules and scoping rules
    1.1. ◦ What is a module?
    1.2. ◦ A first module
    1.3. ◦ The import statement
    1.4. ◦ The module search path
    1.5. ◦ Private names in modules
    1.6. ◦ Library and third-party modules

www.cursoslinux.com.mx
ventas@plct.com.mx
PLCT S.A. de C.V.
Tel.: 55 4522 7839/551800 7696/7224447684

PLCT S.A de C.V

1.7. ◦ Python scoping rules and namespaces

1. . Python programs

1.1. ◦ Creating a very basic program

1.2. ◦ Making a script directly executable on UNIX

1.3. ◦ Scripts on Mac OS X

1.4. ◦ Script execution options in Windows

1.5. ◦ Scripts on Windows vs. scripts on UNIX

1.6. ◦ Programs and modules

1.7. ◦ Distributing Python applications

1. . Using the filesystem

1.1. ◦ Paths and pathnames

1.2. ◦ Getting information about files

1.3. ◦ More filesystem operations

1.4. ◦ Processing all files in a directory subtree

1. . Reading and writing files

1.1. ◦ Opening files and file objects

1.2. ◦ Closing files

1.3. ◦ Opening files in write or other modes

1.4. ◦ Functions to read and write text or binary data

1.5. ◦ Screen input/output and redirection

1.6. ◦ Reading structured binary data with the struct module

1.7. ◦ Pickling objects into files

1.8. ◦ Shelving objects

1. . Exceptions

1.1. ◦ Introduction to exceptions

1.2. ◦ Exceptions in Python

1.3. ◦ Using with

1. . Classes and object-oriented programming

1.1. ◦ Defining classes

1.2. ◦ Instance variables

1.3. ◦ Methods

1. ▪ Class variables

1.1. ◦ Static methods and class methods

1.2. ◦ Inheritance

1.3. ◦ Inheritance with class and instance variables

1.4. ◦ Private variables and private methods

1.5. ◦ Using @property for more flexible instance variables

1.6. ◦ Scoping rules and namespaces for class instances

1.7. ◦ Destructors and memory management

1.8. ◦ Multiple inheritance

www.cursoslinux.com.mx
ventas@plct.com.mx
PLCT S.A. de C.V.
Tel.: 55 4522 7839/551800 7696/7224447684

PLCT S.A de C.V

1. . Graphical user interfaces
    - 1.1. ◦ Installing Tkinter
    - 1.2. ◦ Starting Tk and using Tkinter
    - 1.3. ◦ Principles of Tkinter
    - 1.4. ◦ A simple Tkinter application
    - 1.5. ◦ Creating widgets
    - 1.6. ◦ Widget placement
    - 1.7. ◦ Using classes to manage Tkinter applications
    - 1.8. ◦ What else can Tkinter do?
    - 1.9. ◦ Alternatives to Tkinter
1. . Advanced language features
    - 1.1. ◦ Regular expressions
    - 1.2. ◦ What is a regular expression?
    - 1.3. ◦ Regular expressions with special characters
    - 1.4. ◦ Regular expressions and raw strings
    - 1.5. ◦ Extracting matched text from strings
    - 1.6. ◦ Substituting text with regular expressions
1. . Packages
    - 1.1. ◦ What is a package?
    - 1.2. ◦ A first example
    - 1.3. ◦ A concrete example
    - 1.4. ◦ The __all__ attribute
    - 1.5. ◦ Proper use of packages
1. . Data types as objects
    - 1.1. ◦ Types are objects, too
    - 1.2. ◦ Using types
    - 1.3. ◦ Types and user-defined classes
    - 1.4. ◦ Duck typing
1. . Advanced object-oriented features
    - 1.1. ◦ What is a special method attribute?
    - 1.2. ◦ Making an object behave like a list
    - 1.3. ◦ Giving an object full list capability
    - 1.4. ◦ Subclassing from built-in types
    - 1.5. ◦ When to use special method attributes
    - 1.6. ◦ Metaclasses
    - 1.7. ◦ Abstract base classes
1. . Where can you go from here?
    - 1.1. ◦ Testing your code made easy(-er)
    - 1.2. ◦ Why you need to have tests
    - 1.3. ◦ The assert statement

www.cursoslinux.com.mx
ventas@plct.com.mx
PLCT S.A. de C.V.
Tel.: 55 4522 7839/551800 7696/7224447684

1.4. ◦ Tests in docstrings: doctests

1.5. ◦ Using unit tests to test everything, every time

1. . Moving from Python 2 to Python 3

1.1. ◦ Porting from 2 to 3

1.2. ◦ Testing with Python 2.6 and -3

1.3. ◦ Using 2to3 to convert the code

1.4. ◦ Testing and common problems

1.5. ◦ Using the same code for 2 and 3

1. . Using Python libraries

1.1. ◦ “Batteries included”—the standard library

1.2. ◦ Moving beyond the standard library

1.3. ◦ Adding more Python libraries

1.4. ◦ Installing Python libraries using setup.py

1.5. ◦ PyPI, a.k.a. “the Cheese Shop”

1. . Network, web, and database programming

1.1. ◦ Accessing databases in Python

1.2. ◦ Network programming in Python

1.3. ◦ Creating a Python web application

1.4. ◦ Sample project—creating a message wall