NIMBUS TRAINING
Formez-vous autrement • Tél : 02-40-85-19-32• cedric.tron@nimbus-training.fr

Python-Initiation

Description :

Axée principalement sur le développement de l'autonomie, la formation se déroule sur bpyton et ou ipython (pour ceux qui le connaissent déjà). Tout au long de la formation, les stagiaires font des exercices sur les notions abordées. Un support complet avec les principaux points clefs ainsi que les exercices corrigés est fourni. Enfin la dernière demi-journée est consacrée au codage d'un cas pratique proposé (par le le client lors d'une formation intra entreprise) testé et documenté en pair-programming (ou individuellement) avec revue de code collective

Objectifs

- Connaître les possibilités de Python
- Être capable d'aborder des frameworks Python tels que Django, Flask ou Pyramid

Publics

Cette formation s’adresse à des développeurs

Durée

3 jours

Pré-requis

Connaissance de base en développement.

Programme de cette formation

Maîtriser les types et objets de bases

- strings
- tuples
- listes (in extension)
- dictionnaires (in extension)
- sets (in extension)
- booléens et savoir quels objets sont vrais
- introspection(dir, help)
- gestion de la mémoire, objets muables et immuables (ce qu'il faut savoir)

Structuration du code

- if, for, while, émuler un swicth_case
- les fonctions (paramètres et passage d'arguments)
- portée des variables

Gérer les exceptions et les erreurs

- try except finally
- raise
- Exception
- assert
- traceback
- pdb

Keywords & builtins functions

- with, map, lambda, filter, any, all etc.

Conventions de codage et de nommage

- pep-0008
- pep-0257
- pylint

Gérer les charsets, encodings et unicode

- Comprendre et maîtriser ce qui provoque, peut provoquer des erreurs du type
- UnicodeError: ASCII decoding error: ordinal not in range(128)
- Tout ce qu'un développeur doit savoir sur les charsets, encodings et unicode

Maîtriser le sys.path

- Comprendre le sys.path les modules, et les packages
- virtual_env

Installer un module tiers

- easy_install, pip et zc.buildout
- Créer un package avec un namespace

P.O.O

- Attributs et méthodes
- héritage, surcharge
- type et object (new_style_class et le mro)
- abstract base class
- les metaclass
- duck_typing

La stdlib

- Parcours de quelques modules phares
- argparse
- logging
- configparser
- os
- subprocess
- re (expressions rationnelles)

Présentation des objets et fonctions avancés

- décorateurs
- itérateurs et générateurs
- context managers
- unittest et docstests
- documenter avec sphinx

Codage d'une appli testée et documentée