

# Python pour Data-scientist

Référence : **DELY010**

Durée : **3 jours**

Certification : **Aucune**

## CONNAISSANCES PREALABLES

- Maîtrise de la programmation Python.

## PROFIL DES STAGIAIRES

- Les développeurs en Python.

## OBJECTIFS

- Savoir utiliser les principaux outils de traitement et d'analyse de données pour Python.
- Savoir extraire des données d'un fichier et les manipuler.
- Mettre en place un modèle d'apprentissage simple.

## CERTIFICATION PREPAREE

Aucune

## METHODES PEDAGOGIQUES

- Mise à disposition d'un poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques, de réflexions et de retours d'expérience
- Le suivi de cette formation donne lieu à la signature d'une feuille d'émargement

## FORMATEUR

Consultant-Formateur expert Bigdata

## METHODE D'EVALUATION DES ACQUIS

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation de fin de stage adressée avec la facture

## CONTENU DU COURS

### Positionnement Python

- Besoins des data-scientist:calculs, analyse d'images, machine learning, interface avec les bases de données
- Apports de python:grande variété d'outils, expertise dans le domaine du calcul scientifique
- Tour d'horizon des outils : pandas, agate, bokeh, scikit-learn, pybrain, tensorflow, keras, mxnet, caffe

### Calculs et graphiques

- NumPy : Base du calcul sur des tableaux
- SciPy : Scientific Tools for Python, couche scientifique
- Manipulation de tableaux, fonctions mathématiques
- Représentation graphique avec basemap et matplotlib
- Mise en oeuvre de SciPy/NumPy : manipulation d'images, détection de contours

### Manipulation de données relationnelles

- Pandas : manipulation de tables de données

- Tableaux avec Pandas : indexation, opérations, algèbre relationnelle
- Stockage dans des fichiers : CSV, h5py, netCDF
- Comparaison et performances Pandas / NumPy

### Machine learning et deep learning

- Présentation de TensorFlow, scikit-learn, keras, mxnet, caffe
- TensorFlow : principe de fonctionnement, plateformes supportées, distribution, APIs fournies en standard, modèles d'apprentissage
- Projet scikit-learn : classification, régression, validation de modèles prédictifs
- Démonstrations avec les modèles fournis par scikit-learn
- Positionnement et comparaison avec Keras, mxnet, caffe