

Formation : L'essentiel de Python et de l'objet

Cours pratique - 5j - 35h00 - Réf. PYT

Prix : 3090 € H.T.

★★★★☆ 4,5 / 5

BEST

Clé en main

Nouvelle édition

Blended

Formation éligible au financement Afdas

Python est un langage de programmation multiplateforme permettant le développement d'une grande variété d'applications. Vous en maîtriserez sa syntaxe, ses principaux mécanismes et son paradigme Objet. Vous découvrirez les fonctionnalités de la bibliothèque de modules standards, implémenterez des interfaces graphiques, accéderez aux données d'une base tout en utilisant des outils permettant de tester et d'évaluer la qualité du code produit.

Objectifs pédagogiques

À l'issue de la formation, le participant sera en mesure de :

- ✓ Maîtriser la syntaxe du langage Python
- ✓ Acquérir les notions essentielles de la programmation objet
- ✓ Connaître et mettre en œuvre les différents modules Python
- ✓ Concevoir des interfaces graphiques
- ✓ Mettre en œuvre les outils de test et d'évaluation de la qualité d'un programme Python

Public concerné

Développeurs, ingénieurs, chefs de projets proches du développement.

Prérequis

Connaissances de base en programmation.

Vérifiez que vous avez les prérequis nécessaires pour profiter pleinement de cette formation en faisant [ce test](#).

Modalités d'évaluation

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

PARTICIPANTS

Développeurs, ingénieurs, chefs de projets proches du développement.

PRÉREQUIS

Connaissances de base en programmation.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

Programme de la formation

1 Syntaxe du langage Python

- Les identifiants et les références. Les conventions de codage et les règles de nommage.
- Les blocs, les commentaires.
- Les types de données disponibles.
- Les variables, l'affichage formaté, la portée locale et globale.
- La manipulation des types numériques, la manipulation de chaînes de caractères.
- La manipulation des tableaux dynamiques (liste), des tableaux statiques (tuple) et des dictionnaires.
- L'utilisation des fichiers.
- La structure conditionnelle if/elif/else.
- Les opérateurs logiques et les opérateurs de comparaison.
- Les boucles d'itérations while et for. Interruption d'itérations break/continue.
- La fonction range.
- L'écriture et la documentation de fonctions.
- Les lambda expression.
- Les générateurs.
- La structuration du code en modules.

Travaux pratiques

Installation et prise en main de l'interpréteur Python.

2 Approche Orientée Objet

- Les principes du paradigme Objet.
- La définition d'un objet (état, comportement, identité).
- La notion de classe, d'attributs et de méthodes.
- L'encapsulation des données.
- La communication entre les objets.
- L'héritage, transmission des caractéristiques d'une classe.
- La notion de polymorphisme.
- Association entre classes.
- Les interfaces.
- Les diagrammes de classes, de séquences, d'activités...
- Notion de modèle de conception (Design Pattern).

Travaux pratiques

Mise en œuvre d'ensemble de classes associées entre elles.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les formations pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque formation ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le participant a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse psh-accueil@orsys.fr.

3 Programmation Objet en Python

- Les particularités du modèle Objet de Python.
- L'écriture de classes et leur instanciation.
- Les constructeurs et les destructeurs.
- La protection d'accès des attributs et des méthodes.
- La nécessité du paramètre Self.
- L'héritage simple, l'héritage multiple, le polymorphisme.
- Les notions de visibilité.
- Les méthodes spéciales.
- L'introspection.
- L'implémentation des interfaces.
- Les bonnes pratiques et les modèles de conception courants.
- L'utilisation du mécanisme d'exception pour la gestion des erreurs.

Travaux pratiques

Pratique des différents concepts Objet au travers de l'implantation de l'étude de cas.

4 Utilisation StdLib

- Les arguments passés sur la ligne de commande.
- L'utilisation du moteur d'expressions régulières Python avec le module "re", les caractères spéciaux, les cardinalités.
- La manipulation du système de fichiers.
- Présentation de quelques modules importants de la bibliothèque standard : module "sys", "os", "os.path".
- Empaquetage et installation d'une bibliothèque Python.
- Les accès aux bases de données relationnelles, le fonctionnement de la DB API.

Travaux pratiques

Mise en œuvre de modules Python : expressions régulières, accès à une base de données,

5 Outils QA

- Les outils d'analyse statique de code (Pylint, Pychecker).
- L'analyse des comptes rendus d'analyse (types de messages, avertissements, erreurs).
- Extraction automatique de documentation.
- Le débogueur de Python (exécution pas à pas et analyse post-mortem).
- Le développement piloté par les tests.
- Les modules de tests unitaires Python (Unittest...).
- L'automatisation des tests, l'agrégation de tests.
- Les tests de couverture de code, profiling.

Travaux pratiques

Utilisation de deux outils d'analyse statiques de code pour la vérification d'un code Python.

6 Création IHM

- Les principes de programmation des interfaces graphiques.
- Présentation de différentes bibliothèques graphiques : Tkinter, Qt6, GTK+, wxWidgets.
- Les principaux conteneurs.
- Présentation des widgets disponibles (Button, Radiobutton, Entry, Label, Listbox, Canvas, Menu, Scrollbar, Text...).
- Le gestionnaire de fenêtres.
- Le placement des composants, les différents layouts.
- La gestion des événements, l'objet event.
- Utilisation du Modèle-Vue-Contrôleur dans une IHM.

Travaux pratiques

Conception d'une interface graphique avec la bibliothèque Tkinter.

7 Interfaçage Python/C

- Présentation du module Ctypes.
- Le chargement d'une librairie C.
- Appel d'une fonction.
- Présentation de Cython.
- Compiler un script Python en C : création d'extension compilée.
- Utilisation des annotations pour améliorer les performances de compilation.
- L'utilisation du profileur de code.

Travaux pratiques

Appel de fonctions écrites en C depuis Python. Compilation d'une extension pour Python.

8 Conclusion

- Analyse critique de Python.
- L'évolution du langage.
- Éléments de webographie et de bibliographie.

Parcours certifiants associés

Pour aller plus loin et renforcer votre employabilité, découvrez les parcours certifiants qui contiennent cette formation :

- [Parcours certifiant Concevoir et développer une application informatique en Python - Réf. ZCT](#)
- [Parcours certifiant Développer une application en Python - Réf. KZO](#)

Options

Blended : 290 € HT

Approfondissez les connaissances acquises en formation grâce aux modules e-learning de notre [Chaîne e-learning développement logiciel Python](#). Un apprentissage flexible et complet, à suivre à votre rythme dès le premier jour de votre présentiel.

Solutions de financement

Plusieurs solutions existent pour financer votre formation et dépendent de votre situation professionnelle.

Découvrez-les sur notre page [Comment financer sa formation](#) ou [contactez votre conseiller formation](#).

Financement par les OPCO

- Adhérents Afdas, découvrez les avantages négociés par votre OPCO en cliquant [ici](#)

Horaires

Les cours ont lieu de 9h à 12h30 et de 14h à 17h30.

Les participants sont accueillis à partir de 8h45. Les pauses et déjeuners sont offerts.

Pour les formations de 4 ou 5 jours, quelle que soit la modalité, les sessions se terminent à 16h le dernier jour.

Dates et lieux

CLASSE À DISTANCE

2026 : 23 mars, 20 avr., 18 mai, 8 juin, 22 juin, 27 juil., 24 août, 7 sep., 28 sep., 12 oct., 2 nov., 23 nov., 14 déc.

LYON

2026 : 20 avr., 8 juin, 27 juil., 7 sep., 2 nov.

ANGERS

2026 : 8 juin, 24 août, 2 nov.

BORDEAUX

2026 : 18 mai, 27 juil., 23 nov.

CAEN

2026 : 8 juin, 24 août, 2 nov.

DIJON

2026 : 8 juin, 7 sep., 14 déc.

LAUSANNE

2026 : 18 mai, 24 août, 14 déc.

LILLE

2026 : 18 mai, 27 juil., 23 nov.

MARSEILLE

2026 : 8 juin, 24 août, 2 nov.

MONTPELLIER

2026 : 8 juin, 7 sep., 14 déc.

NANCY

2026 : 8 juin, 24 août, 2 nov.

NIORT

2026 : 8 juin, 27 juil., 12 oct.

PARIS LA DÉFENSE

2026 : 23 mars, 20 avr., 18 mai, 8 juin, 22 juin, 27 juil., 24 août, 7 sep., 28 sep., 12 oct., 2 nov., 23 nov., 14 déc.

AIX-EN-PROVENCE

2026 : 18 mai, 27 juil., 23 nov.

AVIGNON

2026 : 8 juin, 24 août, 2 nov.

BREST

2026 : 8 juin, 24 août, 2 nov.

CLERMONT-FERRAND

2026 : 8 juin, 24 août, 2 nov.

GRENOBLE

2026 : 8 juin, 27 juil., 12 oct.

LE MANS

2026 : 8 juin, 24 août, 2 nov.

LIMOGES

2026 : 8 juin, 24 août, 2 nov.

METZ

2026 : 8 juin, 24 août, 2 nov.

MULHOUSE

2026 : 8 juin, 24 août, 2 nov.

NANTES

2026 : 8 juin, 7 sep., 14 déc.

ORLÉANS

2026 : 8 juin, 27 juil., 12 oct.

PAU

2026 : 8 juin, 24 août, 2 nov.

RENNES

2026 : 8 juin, 7 sep., 14 déc.

SAINT-ETIENNE

2026 : 8 juin, 24 août, 2 nov.

STRASBOURG

2026 : 18 mai, 27 juil., 23 nov.

TOULOUSE

2026 : 18 mai, 27 juil., 23 nov.

VALENCE

2026 : 8 juin, 24 août, 2 nov.

GENÈVE

2026 : 18 mai, 24 août, 14 déc.

REIMS

2026 : 8 juin, 27 juil., 12 oct.

ROUEN

2026 : 8 juin, 27 juil., 12 oct.

SOPHIA-ANTIPOLIS

2026 : 8 juin, 7 sep., 14 déc.

TOULON

2026 : 8 juin, 24 août, 2 nov.

TOURS

2026 : 8 juin, 27 juil., 12 oct.

BRUXELLES

2026 : 8 juin, 7 sep., 23 nov.

LUXEMBOURG

2026 : 8 juin, 7 sep., 23 nov.