Stage pratique de 5 jour(s)
Réf : PYT

Participants

Développeurs, ingénieurs, chefs de projets proches du développement.

Pré-requis

Connaissances de base en programmation.

Prix 2020 : 2890€ HT

Dates des sessions

AIX
16 nov. 2020, 01 mar. 2021
14 juin 2021, 27 sep. 2021

ANGERS
01 fév. 2021, 12 avr. 2021
31 mai 2021, 19 juil. 2021

BORDEAUX
16 nov. 2020, 11 jan. 2021
14 juin 2021, 06 sep. 2021

BRUXELLES
07 déc. 2020, 04 jan. 2021
01 mar. 2021, 12 avr. 2021
31 mai 2021, 05 juil. 2021
30 août. 2021

CLASSE A DISTANCE
19 oct. 2020, 02&16 nov. 2020
07&14 déc. 2020, 11 jan. 2021
01 fév. 2021, 01 mar. 2021
19 avr. 2021, 03 mai 2021
21 juin 2021, 05 juil. 2021
23 août. 2021, 06 sep. 2021

DIJON
23 nov. 2020, 11 jan. 2021
08 mar. 2021, 31 mai 2021
19 juil. 2021, 06 sep. 2021

GENEVE
07 déc. 2020, 01 fév. 2021
12 avr. 2021, 07 juin 2021
09 août. 2021

GRENOBLE
02 nov. 2020, 04 jan. 2021
01 mar. 2021, 12 avr. 2021
31 mai 2021, 05 juil. 2021
30 août. 2021

LILLE
02 nov. 2020, 01 fév. 2021
17 mai 2021, 13 sep. 2021

LIMOGES
16 nov. 2020, 11 jan. 2021
08 mar. 2021, 31 mai 2021
19 juil. 2021, 06 sep. 2021

LUXEMBOURG
07 déc. 2020, 25 jan. 2021
12 avr. 2021, 07 juin 2021
02 août. 2021

LYON
23 nov. 2020, 15 fév. 2021
15 mar. 2021, 26 avr. 2021
03 mai 2021, 14 juin 2021
26 juil. 2021, 13 sep. 2021

MONTPELLIER
23 nov. 2020, 04 jan. 2021
01 mar. 2021, 12 avr. 2021
31 mai 2021, 05 juil. 2021
30 août. 2021

Python, programmation Objet

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 PEDAGOGIQUES

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

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.
- Présentation d'UML.
- Les diagrammes de classes, de séquences, d'activités...
- Notion de modèle de conception (Design Pattern).

Travaux pratiques
Modélisation en UML d'un cas d'étude simple.

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 visibilités.
- Les méthodes spéciales.
- L'introspection.
- L'implémentation des interfaces.
- Les bonnes pratiques et les modèles de conception courants.

NANCY

16 nov. 2020, 11 jan. 2021
12 avr. 2021, 31 mai 2021
19 juil. 2021, 06 sep. 2021

NANTES

02 nov. 2020, 01 fév. 2021
14 juin 2021, 13 sep. 2021

NIORT

01 fév. 2021, 12 avr. 2021
31 mai 2021, 19 juil. 2021

ORLEANS

07 déc. 2020, 01 fév. 2021
01 mar. 2021, 03&31 mai 2021
05 juil. 2021, 30 aoû. 2021

PARIS

19 oct. 2020, 02&16 nov. 2020
07&14 déc. 2020, 11 jan. 2021
01&15 fév. 2021, 01&15&29 mar. 2021
19 avr. 2021, 03&17&31 mai 2021
21 juin 2021, 05&19 juil. 2021
23 aoû. 2021, 06&20 sep. 2021

REIMS

07 déc. 2020, 11 jan. 2021
12 avr. 2021, 31 mai 2021
19 juil. 2021, 06 sep. 2021

RENNES

16 nov. 2020, 04 jan. 2021
01 mar. 2021, 12 avr. 2021
31 mai 2021, 05 juil. 2021
02 aoû. 2021

ROUEN

07 déc. 2020, 11 jan. 2021
12 avr. 2021, 31 mai 2021
19 juil. 2021, 06 sep. 2021

SOPHIA-ANTIPOLIS

01 mar. 2021, 14 juin 2021
27 sep. 2021

STRASBOURG

16 nov. 2020, 01 fév. 2021
17 mai 2021, 13 sep. 2021

TOULON

16 nov. 2020, 11 jan. 2021
12 avr. 2021, 31 mai 2021
19 juil. 2021, 06 sep. 2021

TOULOUSE

16 nov. 2020, 15 mar. 2021
07 juin 2021, 13 sep. 2021

TOURS

16 nov. 2020, 01 fév. 2021
01 mar. 2021, 03&31 mai 2021
05 juil. 2021, 30 aoû. 2021

Modalités d'évaluation

L'évaluation des acquis se fait tout au long de la session au travers des multiples exercices à réaliser (50 à 70% du temps).

Compétences du formateur

- 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 des outils pylint et pychecker pour la vérification d'un code Python. Mise en œuvre de tests unitaires.

6) Création IHM TkInter

- Les principes de programmation des interfaces graphiques.
- Présentation de la bibliothèque TkInter.
- 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.
- Les applications multifenêtres.

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.
- La réécriture d'une fonction Python en C avec l'API Python/C.
- La création de modules C pour Python.
- L'interpréteur Python dans C.
- L'utilisation du profileur de code.

Travaux pratiques
Appel de fonctions écrites en C depuis Python. Création de modules C pour Python avec Pyrex.

8) Conclusion

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

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.

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 stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.

• A l'issue de chaque stage 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 stagiaire a bien assisté à la totalité de la session.