

Développeur Python

Programmation Python en IoT pour Raspberry

5 jours (35h00) | ★★★★★ 4,6/5 | PYT-IOT | Évaluation qualitative de fin de stage |
Formation délivrée en présentiel ou distanciel ⁽¹⁾

Formations Informatique > Langages et développement > Développeur Python



À l'issue de ce stage vous serez capable de :

- Programmer efficacement en Python des applications IoT pour Raspberry Pi.

Niveau requis

Etre initié à la programmation structurée et avoir des connaissances sur l'architecture matérielle d'un PC et de Linux souhaitées.

Public concerné

Programmateurs et chefs de projets.

Cette formation :

- Est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- Bénéficie d'un suivi de son exécution par une feuille de présence émarginée par demi-journée par les stagiaires et le formateur.

(1) Modalité et moyens pédagogique :

Formation délivrée en présentiel ou distanciel * (e-learning, classe virtuelle, présentiel à distance). Le formateur alterne entre méthodes ** démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification.

Les moyens pédagogiques mis en oeuvre (variables suivant les formations) sont : ordinateurs Mac ou PC (sauf pour les cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel). Environnements de formation installés sur les postes de travail ou en ligne. Supports de cours et exercices.

* Nous consulter pour la faisabilité en distanciel. ** Ratio variable selon le cours suivi.

Programme

Jour 1

Présentation de Raspberry

- Architecture interne : RAM, CPU, Socket, Ethernet , GPIO (General Purpose Input/Output)...
- Système d'exploitation Linux : Raspbian, Debian...
- Périphériques et montages électroniques externes pour Raspberry Pi
- IDE (Integrated Development Environment) pour Python : PyDev, Thonny...

Exemples de travaux pratiques (à titre indicatif)

- *Prise en main de Raspberry*
- *Installation et configuration de l'IDE Python*

Administration de Raspberry

- Gestion de sessions
- Gestion de fichiers
- Gestion de processus et logs
- Installation de logiciels
- Image et boot : Berryboot, Noobs et IMG
- Gestion de périphériques /dev et USB
- Système de fichiers : mount, /dev , /var, /home...
- Configuration réseau et gestion de services

Exemples de travaux pratiques (à titre indicatif)

- *Manipulation de fichiers : commandes de base*
- *Manipulation de processus : commandes de base*
- *Démarrage et arrêt du système : init, shutdown...*
- *Installation de logiciels*
- *Configuration réseaux et installation de services*

Jour 2

Langage de programmation Python

- Langage interprété
- Organisation d'un script ou module
- Exécution et passage d'argument en ligne de commande
- Accéder à une aide
- Commentaires et indentations
- Modèle mémoire : head et GC (Garbage Collector)
- Variables : déclarations et portée
- Types de données scalaires : int, str, float...
- Types agrégés : List, tableaux, set, queue, list, dict, chaîne...
- Opérateurs : arithmétiques, logiques, relationnels, spéciaux...
- Instructions de contrôle : if / else, while, for...
- Notions des fonctions et passage d'arguments
- Gestion des exceptions : try / except / final

Exemples de travaux pratiques (à titre indicatif)

- *Création d'un premier programme Python, avec passages d'arguments*
- *Utilisation des variables et de ses types*
- *Exploitation des tableaux*
- *Utilisation des structures de contrôle principales de Python*
- *Création de fonctions et appel de celles-ci*

- Ajout de la gestion des exceptions aux programmes réalisés durant la journée

Jour 3

Langage de programmation Python - Suite

- Création et utilisation de modules en Python
- Les expressions régulières en Python
- Gestion de modules (pip) : installation, création et importation
- Modules standards : OS, sys, random, string...
- Gestion de I/O : entrées / sorties standards, accès aux fichiers et redirections
- Particularités de la programmation Python sur Raspberry
- Programmation des GPIO (General Purpose Input/Output) Raspberry
- L'instruction "with" en Python
- Les arguments *args, **kargs

Exemples de travaux pratiques (à titre indicatif)

- Création et utilisation de modules
- Utilisation des modules standards
- Accès aux fichiers depuis Python
- Pilotage par Python de montage électronique via GPIO

Aspect avancés du développement Python sur Raspberry

- Traitements parallèles et concurrents en Python : les différences
- Traitements parallèles avec les threads en Python
- La programmation concurrente en Python
- Communication réseaux avec les sockets : TCP (Transmission Control Protocol) et UDP (User Datagram Protocol)
- Les WebSockets en Python et avantage des messages non sollicités

Exemples de travaux pratiques (à titre indicatif)

- Mise en oeuvre du multi-threads en Python
- Mise en oeuvre de la communication via Sockets

Jour 4

La programmation orientée objet (POO)

- Les classes en Python
- Attributs et méthodes
- Les constructeurs
- Les instances
- Le paramètre self
- Héritage en Python
- Utilisation de super en Python
- Mise en oeuvre du polymorphisme
- Les annotations
- Les classe d'exceptions
- La persistance avec pickle et json

Exemples de travaux pratiques (à titre indicatif)

- Mise en oeuvre de la programmation objet en Python
- Création de classes, propriétés et méthodes
- Ajout de constructeurs
- Développement de l'héritage
- Exploitation du polymorphisme
- Ajout du traitement d'exception dans les classes

Conception d'interfaces graphiques

- Présentation de l'API de gestion graphique Tkinter
- Gestion
 - Fenêtres
 - Evènements souris / clavier
 - Gadgets visuels
 - Menus
 - Boîtes de dialogues
 - Texte
 - Graphique vectoriel
- Problématique dite graphique et multi-thread

Exemples de travaux pratiques (à titre indicatif)

- Réalisation d'une interface graphique avec Tkinter
- Intégration à l'interface des différents contrôles visuels et gestion programmatique de ceux-ci

Jour 5

Mise en oeuvre de Raspberry

- Rappels sur GPIO
- Affichage sur Raspberry
- Présentation de PWM (Pulse Width Modulation)
- Présentation de MQTT (Message Queuing Telemetry Transport)

Exemples de travaux pratiques (à titre indicatif)

- *Projet développé d'accès au GPIO depuis un programme Python*
- *Projet d'affichage de nombres aléatoires sur Raspberry*
- *Projet de contrôle de moteurs connectés à Raspberry, via PWM (Pulse Width Modulation)*
- *Projet d'échanges entre deux Raspberry, via le protocole MQTT (Message Queuing Telemetry Transport)*

Modalités d'évaluation des acquis

L'évaluation des acquis se fait :

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation ou une certification (M2i ou éditeur)