

Détail des programmes de formation Hard Skills

Fondamentaux du développement (10 jours, 70h) :

1. Architecture des ordinateurs
2. Notion de réseau informatique et fonctionnement du web
3. Fondamentaux de la programmation (langage de programmation, compilation / interprétation, etc.)
4. Appréhender le développement web
5. Étude des bases du HTML/CSS
6. Notion d'algorithmique
7. Apprentissage d'un langage de programmation : Base du langage Java
8. Appréhender la programmation orienté objet : POO avec Java
9. Comprendre les bases de données et le SQL
10. Utilisation d'un SGBD
11. Conception, modélisation et indispensables d'UML
12. Outils et démarches de la conduite de projet

WordPress (5jours, 35h) :

1. Les fondamentaux du web et les CMS
2. Installations et prise en main de WP
3. Gestion d'un site avec WP (contenu, page, article)
4. Gestion des médias (images, sons et vidéos)
5. Gestion des utilisateurs
6. Emploi de thème et personnalisation
7. Utilisation de widget et de plugin
8. E-commerce avec WP
9. Référencement (SEO), statistique et analyse (Google Analytics)
10. Utilisation avancée de WP (hébergements, préprod, sauvegardes, mises à jour, maintenances et sécurités : niveau de droit, trolling et spamming)
11. Programmations (Hooks, création de plugins, widgets et thèmes)

UX/UI Design (5jours, 35h) :

1. Principe et fondamentaux de l'UX/UI Design
2. Le processus Design
3. Les principaux méthodes et outils
4. Appréhender les utilisateurs
5. Analyse UX design
6. Conception UX/UI design
7. Test et évaluation
8. Étude de cas (maquettage et prototypage)
9. Bonnes pratiques

Développement web front-end (Angular) (7jours-49h) :

1. Découvrir le Framework Angular
2. TypeScript et ES2015
3. Comprendre les bases du Framework d'Angular
4. Les composants, les directives, les modules et les pipes
5. Les services et la communication inter-composants (services, injection de dépendances, et router)
6. Programmation réactive et communication serveur (observables, échange avec un serveur, formulaire)
7. Formulaires avancés (reactiveForms, FormBuilder, contrôles avancés)
8. Authentification et autorisation
9. Les Progressive Web Apps
10. La gestion de l'état applicatif (NgRx / NGXS)
11. Utilisation des tests (tests unitaires et fonctionnels)
12. Les nouveautés d'Angular et concepts avancés
13. Performance et bonne pratique

Développement web back-end (Spring Boot avec kotlin) (7jours-49h) :

1. Introduction à Spring Boot
2. Codage avec Kotlin
3. Concept fondamentaux
4. Gestion des dépendances et de l'auto-configuration
5. Persistance avec JPA et Spring Data
6. Spring Boot et Spring MVC
7. REST dans Spring Boot
8. Journalisation et traces
9. Sécurité et Spring Boot
10. Documentation
11. Spring Boot et le tests (test automatisés, tests unitaires et tests d'intégration)
12. Mise en production

Développement d'application Android (8jours-56h):

1. Généralité sur le développement d'application
2. L'écosystème et architecture Android
3. Présentation, installation et configuration de l'environnement de développement
4. Les interfaces graphiques (UI, widgets, layout, événements, adapter, adapter view et fragments)
5. Les Intents
6. Formulaire et listes
7. Persistance et partage
8. Programmation asynchrone et Réseautique
9. Utilisation des connectiques
10. Android Wear
11. Compléments et bonne pratique

Développement d'application IOS (8jours-56h) :

1. Écosystème du développement pour équipements mobiles d'Apple
2. Mise en place de l'environnement
3. Le langage Swift
4. Les notions fondamentales
5. Construction d'interface graphique
6. Navigation et formulaires
7. Lecture, enregistrement et échange de donnée
8. Asynchronisme et tâches de fond (Multitâche et background)
9. Responsive design (Swift UI et UIKit)
10. Le Core Motion et les capteurs
11. Archivage et publication

Développement Full Stack (JavaScript) (10j-70h) :

1. Les fondamentaux du web
2. HTML
3. CSS
4. JavaScript
5. Document Object Model
6. SQL et SGBD
7. React.js
8. Node.js
9. React Native

Professionalisation (3j-21h) :

1. Cycle de vie d'un logiciel
2. Les méthodologies de développement
3. De l'importance de la conception
4. Règle de codage (organisation et indentation du code, conseils de nommage, usage des commentaires, gestion des espaces et des sauts de ligne, etc.)
5. Amélioration continue et routine de codage (codage, test, refactoring, analyse, etc.)
6. Documentation, collaboration et versionning
7. Automatisation (DevOps) et virtualisation
8. Internationalisation
9. Les tests (tests unitaires, tests d'intégrations, etc.)
10. Performances et sécurité
11. Profiter des richesses de l'internet