

Développement Front End - Angular

Référence : OPS006

Durée : 5 jours

Certification : Aucune

CONNAISSANCES PREALABLES

- Maîtrise de HTML et CSS.

PROFIL DES STAGIAIRES

- Chefs de projet. • Développeurs. • Webmaster.

OBJECTIFS

- Maîtriser la syntaxe du langage Javascript. • Manipuler la structure DOM d'une page HTML. • Gérer la programmation événementielle. • Maîtriser les concepts du framework Angular. • Utiliser Angular pour développer des applications Web performante .

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 Développement Internet

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

Concepts de base du JavaScript

- Principes et fonctionnements
- Variables et types de données
- Opérateurs
- Instructions
- Structures de contrôles (conditions, boucles...)
- Définition et utilisation de fonctions
- Objets prédéfinis (document, window ...)
- Contrôle de saisies et traitement de formulaires
- Gestion des fenêtres (ouverture, fermeture, communication)
- Boîtes de dialogue
- Atelier : apprendre à valider un formulaire

ANGULAR 6 - Introduction

- Développement d'applications riches
- Panorama des frameworks disponibles

- Présentation de l'environnement de développement NodeJS
- Atelier : Installation et mise en route d'un projet Angular

ES2015

- Lien entre Angular et ES2015/TS
- Classes, fondamentaux: constructeur, méthode et gestion de contexte
- Classes, avancés: héritage, méthodes statiques
- Opérateurs set et get
- Littéraux de Gabarits
- Fonctions fléchées
- Variables de bloc

Typescript

- Déclaration de type
- Interface et formes
- Décorateurs

Première application

- Dépendances d'Angular
- Déclarer un composant
- Ajouter un gabarit
- Lancer l'application avec un serveur NodeJS
- Re-compiler automatiquement à chaque modification: le veilleur
- Ajouter des données, créer ses classes de modèle
- Lier une donnée au gabarit, expressions et opérateur «moustache»
- Gérer des collections, directive *ngFor
- Atelier : Création d'une première application

Formulaire

- Liaison des événements d'entrée
- L'objet \$event
- Variable de référence de gabarit
- Création de formulaire avec la syntaxe de gabarit
- Liaison de données à double sens
- Suivre et valider les saisies utilisateur avec ngControl
- Soumettre le formulaire avec ngSubmit
- Atelier : Gestion des entrées utilisateurs

Injection de dépendances

- Principes de DI
- Configurer l'injecteur
- Enregistrer des fournisseurs dans un composant
- Création de l'injecteur
- Différents types de fournisseurs : classe, valeur, fabrique...
- Atelier : Architecturer notre application sous forme d'une organisation de services

La Syntaxe de gabarit en détail

- Interpolation, expression et instruction
- Syntaxe de liaison : sens unique, double sens, cibles de liaison
- Directives natives : NgClass, NgStyle, NgIf, NgSwitch, NgFor
- Propriétés d'entrée et de sortie
- Opérateurs d'expression de gabarit «|» et «?.»

Le client HTTP

- Principe de communication asynchrone
- Récupérer des données avec http.get
- Observable RxJS
- Prise en charge du transport JSON
- Gestion des erreurs
- Envoi de données au serveur
- Fonctionnement des promesses
- Sources externes: CORS
- Atelier : Mettre en place un client d'API ReST

Routage

- Routage côté client, hashtag et HTML5
- Configuration des routes sur un composant
- Récupérer des paramètres de routes
- Stratégie de routes
- Routes imbriquées
- Atelier : Mettre en place une application multi-vues

Tester l'application

- Fondamentaux des tests avec Jasmine
- Automatisation des tests avec Karma
- Tester des composants simples
- Tester des composants avec services asynchrones
- Tester des formulaires
- Atelier : Multiples exemples de tests