

Stage pratique de 3 jour(s)  
Réf : JVS

## Participants

Cette formation s'adresse à tout développeur, chef de projet ou intégrateur Web souhaitant optimiser les temps de développement par la maîtrise des utilitaires modernes.

## Pré-requis

Bonnes connaissances du HTML et des technologies du Web. Connaissance du langage de programmation JavaScript.

## Dates des sessions

### PARIS

09 mar. 2020, 22 juin 2020  
28 sep. 2020

## 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

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

# Industrialisation Front-End, maîtriser ses pratiques de développement

*Depuis quelque temps, le langage JavaScript, en plein essor, a modifié l'environnement du développeur Front-End. Ce stage vous présentera les outils modernes qui vous permettront de créer un Workflow efficace, automatiser les tâches redondantes et produire des applications modulaires.*

## OBJECTIFS PEDAGOGIQUES

Maîtriser les "SuperSet" HTML CSS JavaScript  
Automatiser les tâches de développement  
Organiser son Workflow  
Intégrer efficacement les contributions

### 1) Composants d'une application Web

### 2) Choisir l'environnement de développement

### 3) "Versionner" avec GIT

### 4) Node.js comme utilitaire de développement

### 5) Automatisation des tâches avec Gulp

### 6) Intégration des tests automatisés

### 7) Créer un template de projet

## Méthodes pédagogiques

*Développement, présentation et mise en oeuvre d'applications existantes.*

## Travaux pratiques

*Vous configurerez pour chaque étape de vos développements les outils d'optimisation afin de pouvoir au final générer vos profils de projets.*

## 1) Composants d'une application Web

- Structurer un projet d'application Web.
- HTML5, les frameworks (bootstrap, foundation...).
- CSS3, automatisation avec LESS ou SASS.
- JavaScript, normaliser le code en modules (API CommonJS et AMD).
- "SuperSet JavaScript" : simplifier le développement avec TypeScript.
- Organiser le développement JavaScript.
- Optimiser l'empreinte des ressources.

### Travaux pratiques

*Migrer une application Web en utilisant le développement modulaire JavaScript. Développer un Framework CSS avec SASS.*

## 2) Choisir l'environnement de développement

- Offre du marché (SublimeText, Eclipse...).
- Choisir et configurer son IDE (extensions...).
- Quelles fonctionnalités attendre, "live Template", "code snippet" ?.
- Les solutions pour développer en ligne (Cloud 9...).
- Les environnements pré-intégrés (PrePros).

### Travaux pratiques

*Découverte des différentes plates-formes. Personnaliser l'IDE SublimeText.*

## 3) "Versionner" avec GIT

- Gestion de source : centralisée versus distribuée.
- Concepts de base, initialiser un repository.
- Organiser les branches de développement.

### Travaux pratiques

*Mise en place d'un processus de développement en branches.*

## 4) Node.js comme utilitaire de développement

- Présentation de Node.js. Le Node Package Manager
- Les modules indispensables pour le développeur.
- Npm et Bower, objectifs et confusion.
- Gestion des dépendances. Créer un package npm.

### Travaux pratiques

*Créer un module pour configurer les dépendances du projet.*

## 5) Automatisation des tâches avec Gulp

- Présentation : automatisation des tâches.

est ensuite analysé par nos équipes pédagogiques.

- Une feuille d'émergence 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.

- Définition des types de tâches : créer une commande.
- Structure d'un fichier gulpfile.js.
- Les contributions indispensables.
- Créer un "Build" efficace avec Gulp.

#### **Travaux pratiques**

*Configurer et utiliser Gulp.*

### 6) Intégration des tests automatisés

- Utilitaires de test : Jasmine, Mocha...
- Recommandation pour l'écriture des tests unitaires.
- Karma : présentation et configuration.
- Générer les rapports de test et de couverture du code.
- End to End Testing : interface utilisateur.

#### **Travaux pratiques**

*Tests unitaires avec Jasmine. Automatisation avec Karma et Istanbul.*

### 7) Créer un template de projet

- Présentation, avantage et inconvénient du scaffolding.
- Quels utilitaires : Yeoman, Lineman ?
- Les générateurs de Yo. Création de générateur.