

# Formation : TypeScript - Produire ses développements JavaScript

Formation pratique - 2j - 14h00 - Réf. TSR

Prix : 1610 CHF H.T.

★★★★☆ 4,7 / 5

Vous découvrirez le langage TypeScript, la surcouche JavaScript pour les développements Front-End. Vous comprendrez le typage, vous maîtriserez les classes et l'héritage pour la programmation orientée objet. Vous gagnerez en productivité afin de mettre en place des interfaces maintenables.

## Objectifs pédagogiques

À l'issue de la formation, le participant sera en mesure de :

- ✓ Prendre en main les outils de développement
- ✓ Créer des fonctions et des tableaux
- ✓ Maîtriser l'héritage avec TypeScript
- ✓ Assembler les codes pour le serveur

## Public concerné

Développeurs et chefs de projets web.

## Prérequis

Programmeurs ayant des connaissances JavaScript de base.

Vérifiez que vous avez les prérequis nécessaires pour profiter pleinement de cette formation en faisant [ce test](#).

## Modalités d'évaluation

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Programme de la formation

### PARTICIPANTS

Développeurs et chefs de projets web.

### PRÉREQUIS

Programmeurs ayant des connaissances JavaScript de base.

### 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.

### MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## 1 Le principe de TypeScript

- Pourquoi Microsoft a créé cette alternative ?
- La surcouche JavaScript.
- La comparaison entre les 2 JavaScript et TypeScript.
- Le choix de l'éditeur, des plug-ins et l'installation.
- L'écosystème d'outils.
- La configuration de l'environnement de développement.
- La transpilation débogage.

### Travaux pratiques

Installation de VS Code et mise en place des plug-ins.

## 2 Configurer le compilateur.

- Le fichier "tsconfig.json".
- Les options du compilateur.
- Créer un "build".
- Les stratégies de projets à configuration multiples.
- L'inclusion et l'exclusion de ressources.
- La création d'un workflow personnalisé.

### Travaux pratiques

Manipulation du compilateur TypeScript et conception d'un build.

## 3 TypeScript Basics & Basic Types

- Les types TypeScript versus JavaScript.
- Quand et quoi typer ?
- L'expression statique de types.
- Assignation et inférence.
- Les tuples, les enums, les aliases (type personnalisés).
- Les cas de typage particulier (never, void...).
- Les fonctions et les tableaux.

### Travaux pratiques

Création de variables de classe avec le typage standard et particulier.

## 4 Next-generation JavaScript

- L'impact sur la syntaxe JavaScript.
- Automatiser la documentation.
- Utiliser TypeScript avec du code JavaScript standard.
- Inclure/générer des fichiers de définition pour la compatibilité.
- Le bénéfice des "getter,setter" implicites.
- La programmation "Ahead-of-Time", réduire la dette technique.
- Les modules et les espaces de noms.

### Travaux pratiques

Création d'un programme avec TypeScript en combinaison avec JavaScript.

### 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 formations pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque formation 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 participant a bien assisté à la totalité de la session.

### MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

### ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse [psh-accueil@orsys.fr](mailto:psh-accueil@orsys.fr).

## 5 Améliorer la POO avec TypeScript

- Les rappels des Pattern POO : Singleton, Factory...
- L'usage des interfaces TypeScript "Duck Typing".
- Les classes & les interfaces.
- Le modificateur de classe : "public, private, readonly..."
- Les méthodes: "private, protected, overrides..."
- L'héritage simple, multiple.

### Travaux pratiques

Conception d'héritage multiple au sein des classes précédemment créées.

## 6 Advanced types

- La gestion des types personnalisés.
- Les types génériques et les inférences.
- Les classes génériques.
- Les restrictions des types génériques.

### Travaux pratiques

Développement et mise en place de types personnalisés.

## 7 Décorateurs

- Les décorateurs et les "MetaData".
- Comprendre les "factories" de décorateurs.
- Les décorateurs de classe, propriété, méthode, paramètres.
- Créer des décorateurs personnalisés.
- Implémenter les abstractions utiles.

### Travaux pratiques

Implémentation d'un décorateur de classes.

## 8 TypeScript en pratique

- Usage avec Node.js.
- Livrer du code JavaScript compatible avec TypeScript.

## Dates et lieux

### CLASSE À DISTANCE

2026 : 8 juin, 21 sep., 26 nov.