DÉVELOPPEMENT FRONTEND

TypeScript pour les développeurs JavaScript

Ajoutez TypeScript à vos projets JavaScript et React : typage strict, interfaces et unions, génériques, types utilitaires et avancés, composants et hooks typés, migration progressive.

DURÉE	TARIF	NIVEAU	LANGUE	GROUPE	FORMAT
3j (21h)	**1 275 € / participant** Inter-entreprise. Intra : sur devis, au plus 1 275 € / participant	**Intermédiaire**	**Français**	**3-10**	**Formation** Présentiel ou à distance

1 PUBLIC VISÉ

- Développeurs JavaScript qui veulent fiabiliser leur code avec le typage statique
- Développeurs React qui veulent typer leurs composants, leurs props et leurs hooks

2 PRÉREQUIS

- Écrire du JavaScript moderne : fonctions fléchées, déstructuration, modules, classes
- Créer un composant React avec des props et des hooks
- Aucune connaissance de TypeScript n’est requise

3 OBJECTIFS PÉDAGOGIQUES

- Configurer TypeScript dans un projet (tsconfig strict) et typer variables, fonctions, objets et tableaux
- Modéliser des données avec interfaces, alias de type, unions discriminées et types littéraux, et réduire les types par narrowing
- Écrire des fonctions, des classes et des types génériques avec contraintes
- Transformer des types avec les utility types, keyof, les types conditionnels et les types mappés
- Typer les props, les événements, les hooks et le contexte d’une application React
- Migrer progressivement un projet JavaScript vers TypeScript sans recourir à any

4 PROGRAMME DÉTAILLÉ

Jour 1 (7h) : Les bases de TypeScript

Module 1 : Mettre en place TypeScript et typer les bases (3h30)

Pourquoi TypeScript et l’environnement (1h30)

- Les limites de JavaScript sur un gros projet ; ce qu’apporte le typage statique
- TypeScript 6 et 7 : le compilateur réécrit en Go, beaucoup plus rapide
- tsc et tsconfig.json : le mode strict, activé par défaut depuis TypeScript 6
- Exécuter du TypeScript : Node.js (qui retire lui-même les types) ou tsx ; TypeScript dans un projet Vite
- Exercice d’entraînement : créer un projet et lire ses premières erreurs de compilation

Types de base, fonctions, objets et tableaux (2h)

- Types de base ; any et unknown ; never ; inférence
- Fonctions : paramètres, retour, paramètres optionnels, surcharges
- Objets : propriétés optionnelles et readonly ; tableaux et tuples
- TP évalué : bibliothèque utilitaire entièrement typée et entité utilisateur avec propriétés obligatoires et optionnelles

Module 2 : Interfaces, unions et narrowing (3h30)

Interfaces et alias de type (1h30)

- Interface ou alias de type : la règle de choix
- extends et intersections
- Exercice d’entraînement : entités d’une boutique en ligne

Unions, littéraux et narrowing (2h)

- Unions et unions discriminées
- Types littéraux, as const, satisfies ; enums et leurs alternatives
- Narrowing : typeof, in, fonctions de garde ; vérification d’exhaustivité
- TP évalué : système de notifications par unions discriminées et statuts de commande en types littéraux

Jour 2 (7h) : Génériques et types avancés

Module 3 : Génériques (3h30)

Comprendre les génériques (1h30)

- Le problème résolu : réutiliser sans perdre le typage
- Fonctions et types génériques ; contraintes avec extends
- Exercice d’entraînement : fonction de tri générique

Génériques avancés (2h)

- Interfaces et classes génériques ; valeurs par défaut
- keyof et types indexés
- TP évalué : enveloppe de réponse d’API générique et repository générique typé

Module 4 : Utility types et types avancés (3h30)

Utility types (1h30)

- Partial, Required, Readonly, Pick, Omit, Record
- Exclude, Extract, NonNullable ; ReturnType, Parameters
- Exercice d’entraînement : type d’un formulaire de mise à jour partielle

Types conditionnels et mappés (2h)

- Types conditionnels et infer
- Types mappés ; types gabarits (template literal types)
- TP évalué : typer la configuration d’une API à partir d’un seul type source, sans duplication

Jour 3 (7h) : TypeScript avec React et projet de synthèse

Module 5 : TypeScript avec React (3h)

Composants typés (1h30)

- Typer les props ; typage direct plutôt que React.FC
- children : ReactNode ; la ref passée comme une prop (React 19)

- Événements : MouseEvent, ChangeEvent, FormEvent
- Exercice d’entraînement : bouton générique à variantes typées

Hooks et contexte (1h30)

- useState, useRef, useReducer avec unions discriminées
- useContext : un contexte fortement typé
- Hooks personnalisés génériques
- TP évalué : hook useForm générique et contexte typé

Module 6 : Migration : projet de synthèse, évalué (4h)

Stratégie de migration (1h)

- Migrer fichier par fichier : allowJs, checkJs
- Fichiers de déclaration .d.ts ; installer les types avec @types
- Les règles strictes qui comptent : noImplicitAny, strictNullChecks
- Exercice d’entraînement : migrer un composant et corriger ses erreurs de type

Projet de synthèse (3h)

- Projet de synthèse, réalisé individuellement : migrer vers TypeScript strict une application React existante (composants, hooks personnalisés, contexte, appels d’API), sans any
- Retour individuel du formateur sur le projet rendu

Le contenu peut être ajusté selon le niveau, les prérequis et les besoins des participants.

5 COMPÉTENCES VISÉES

- Écrire du TypeScript strict et modéliser des données
- Écrire des types réutilisables : génériques et types utilitaires
- Typer une application React
- Migrer un projet JavaScript vers TypeScript

6 MODALITÉS PÉDAGOGIQUES

Formation délivrée en présentiel ou à distance, en classe virtuelle. Le formateur alterne entre méthode démonstrative, méthode interrogative (analyse d’erreurs TypeScript réelles et des choix de typage) et méthode active (exercices d’entraînement, TP évalués et projet de synthèse).

7 MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Support de cours numérique mis à disposition des apprenants
- Dépôt GitHub de démonstration avec exercices et corrections par module
- Environnement de développement : VS Code + Node.js en version LTS + TypeScript + Vite
- Pour la classe virtuelle : visioconférence (Microsoft Teams ou équivalent), partage d’écran, chat en direct
- Supports remis par e-mail et sur la plateforme pédagogique LaPolaris (ressources, émargement)

En formation intra-entreprise sur le site du client, celui-ci s’assure et s’engage à disposer de toutes les ressources matérielles nécessaires à son bon déroulement (salle, postes de travail, logiciels, accès à Internet), conformément aux prérequis indiqués dans le programme et aux moyens prévus dans la convention.

8 MODALITÉS D'ÉVALUATION

- Questionnaire de positionnement individuel renseigné par chaque participant avant le démarrage de la formation, analysé par le formateur et donnant lieu, le cas échéant, à un ajustement du déroulé
- Exercices d’entraînement réalisés par chaque participant sur son poste, corrigés en séance
- Évaluation des acquis par les TP évalués, réalisés individuellement sur poste : un par module, puis le projet de synthèse décrit dans le dernier module du programme ; chaque objectif est évalué dans au moins un TP
- Grille de suivi renseignée par le formateur pour chaque participant et pour chacun des objectifs de l’action, au regard des critères d’évaluation ci-dessous
- Questionnaire d’appréciation renseigné par les participants en clôture de formation, et enquête à froid dans un délai de trois à six mois

9 CRITÈRES D'ÉVALUATION

- tsconfig en mode strict ; projet qui compile sans erreur ; pas d’annotation inutile là où l’inférence suffit
- Interface ou alias choisi selon le besoin ; union discriminée traitée dans tous ses cas, avec vérification d’exhaustivité ; narrowing sans conversion forcée
- Génériques contraints qui évitent la duplication ; types déduits à l’appel sans annotation
- Types dérivés d’un type source avec les utility types, keyof, types conditionnels ou mappés ; aucun type recopié à la main
- Props, événements, hooks et contexte typés ; aucun any ; erreurs de type signalées dans l’éditeur
- Projet migré qui compile en mode strict ; aucun any ni @ts-ignore ; types des bibliothèques installés

10 INDICATEURS DE RÉSULTATS

Les indicateurs de résultats de cette formation (taux de satisfaction et taux de réussite) seront publiés sur sa fiche en ligne à l’issue des premières sessions réalisées.

11 MODALITÉS DE VALIDATION

Une attestation de fin de formation est remise à chaque participant à l’issue du parcours. Elle mentionne **l’intitulé** et la **nature de l’action**, sa **durée** en heures, sa **période de réalisation**, **les objectifs de l’action** et le **niveau d’atteinte** de chacun d’eux, exprimé en trois valeurs : **atteint**, **partiellement atteint**, **non atteint**. **Une attestation d’assiduité**, établie à partir des émargements, est délivrée dans les mêmes conditions.

Formation non certifiante : elle ne délivre pas de certification enregistrée au RNCP ou au Répertoire spécifique.

12 SUIVI ET ACCOMPAGNEMENT

- Émargement électronique signé par chaque participant et par le formateur à chaque demi-journée, en présentiel comme à distance
- Traçabilité des activités pédagogiques réalisées
- Attestation d’assiduité délivrée en fin de formation
- Suivi individuel via les QCM de fin de journée, les exercices corrigés et les TP évalués

13 CONDITIONS D'ACCÈS

Formation accessible sans prérequis administratif particulier. Cette formation n’est pas éligible aux financements publics ou mutualisés, notamment au compte personnel de formation et à une prise en charge par un opérateur de compétences.

14 DÉLAIS D'ACCÈS

Inscription possible jusqu’à 15 jours ouvrés avant le début de la session pour permettre la préparation logistique et, si besoin, l’adaptation du contenu au contexte client. Pour toute demande spécifique, nous contacter directement.

ACCESSIBILITÉ · HANDICAP

Nos formations peuvent être accessibles aux personnes en situation de handicap. La question d’un besoin d’aménagement (matériel, temporel ou pédagogique) est posée lors de l’analyse du besoin et du questionnaire de positionnement ; vous pouvez aussi nous le signaler à tout moment, y compris pendant la formation, afin d’étudier les adaptations possibles. Référent handicap : contact@lapolaris.fr

LaPolaris

TÉL. +33762584798

EMAIL contact@lapolaris.fr

WEB lapolaris.fr