

TYPESCRIPT

Durée	2 jours	Référence Formation	4-JS-TYSC
--------------	----------------	----------------------------	------------------

Objectifs

Grace à cette formation vous pourrez
Identifier et développer en TypeScript.
Connaître et utiliser les particularités de TypeScript
Écrire une application en TypeScript
Exploiter l'écosystème de TypeScript
Identifier les meilleurs IDE

Participants

Développeurs Web pour des applications Web responsives en TypeScript.

Pré-requis

Il est essentiel de maîtriser JavaScript

Moyens pédagogiques

Réflexion de groupe et apports théoriques du formateur
Travail d'échange avec les participants sous forme de réunion-discussion
Utilisation de cas concrets issus de l'expérience professionnelle
Validation des acquis par des questionnaires, des tests d'évaluation, des mises en situation et des jeux pédagogiques.
Remise d'un support de cours.

PROGRAMME

PRÉSENTATION DE TYPESCRIPT

- Objectifs du langage TypeScript : les grosses applications
- Histoire de TypeScript, développé par Microsoft
- TypeScript, un sur-ensemble de JavaScript
- Compatibilité JavaScript et TypeScript
- Différences entre les deux langages
- Typage avec TypeScript / L'objet avec TypeScript

PLATEFORME DE DÉVELOPPEMENT

- IDE en présence
- Outils Chrome
- Focus sur Visual Studio Code
- Transpilation avec VS Code
- Assemblage de code pour le navigateur et pour le serveur
- Le .map, mapper entre TypeScript et le JavaScript généré

TYPAGE EN TYPESCRIPT

- Utilisation des types dans TypeScript
- Définition de l'inférence de type
- Types existants dans TypeScript
- Fonctions
- Tableaux (Array)

CAP ÉLAN FORMATION

www.capelanformation.fr - Tél : 04.86.01.20.50

Mail : contact@capelanformation.fr

Organisme enregistré sous le N° 76 34 0908834

version 2024

- Autres types comme Enum...

OBJET AVEC TYPESCRIPT

- Classes dans TypeScript
- Constructeurs, méthodes et propriétés
- Hériter en TypeScript
- Interfaces
- Généricité

MODULES EN TYPESCRIPT

- Deux types de modules
- Les "mixins"
- Faire le "merge de classes" avec les "mixins"
- Type définition avec TypeScript
- Écrire ses propres fichiers de définition