

# Formation : Apprendre à programmer (avec JavaScript)

Novice en développement initiez-vous aux bases de l'algorithme

**Cours pratique - 3j - 21h00 - Réf. IJV**

**Prix : 1940 CHF H.T.**

 4,4 / 5

Ce cours initie à la programmation classique (des variables, des structures de contrôle et de données) et présente sommairement la programmation objet. Les nombreux programmes que vous y réalisez sont en JavaScript, mais les concepts et les méthodes que vous acquérez restent valides quels que soient les langages de vos futures missions. Nous vous proposons de découvrir l'algorithme via le langage JavaScript.

## Objectifs pédagogiques

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

- ✓ Structurer des programmes selon un algorithme
- ✓ Maîtriser les éléments de lexique et de syntaxe d'un langage pour écrire un programme
- ✓ Comprendre les notions et différences des langages compilés et interprétés.
- ✓ Déboguer et tester un programme
- ✓ Accéder à des données et les manipuler
- ✓ Comprendre les grands principes de la Programmation Orientée Objet

## Public concerné

Toute personne devant apprendre à programmer.

## Prérequis

Aucune connaissance particulière.

## Méthodes et moyens pédagogiques

Ce cours contient plus de 60 % de travaux pratiques effectués selon les besoins en Java (cours réf INJ), en C# (cours réf OGR), en Python (cours réf IPY), ou autres déclinaisons de langages..

## PARTICIPANTS

Toute personne devant apprendre à programmer.

## PRÉREQUIS

Aucune connaissance particulière.

## 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.

## 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

### 1 Les fondements de la programmation et la genèse d'un programme

- Qu'est-ce qu'un programme ? Qu'est-ce qu'un langage ? Les différents paradigmes. Quel langage pour quelle application ?
- Les responsabilités d'un programmeur.
- Langages compilés – Langages interprétés
- Qu'est-ce qu'un algorithme ? Les besoins auxquels répond un algorithme. Le concept de pseudo-langage.
- Qu'est-ce qu'une librairie, bibliothèque / un Framework ? Leur rôle, leur usage.

#### Travaux pratiques

Présentation de différents langages (Java, C#, C, C++, Python). Ecriture d'un premier algorithme en pseudo-langage.

### 2 L'algorithmique dans l'environnement JavaScript

- Création des fichiers de base, les outils.
- Utilisation des commentaires. Pourquoi commenter les développements ?
- Affichage : document, console de debug, boîte de dialogue, boîte d'alerte.

#### Travaux pratiques

Découverte de l'environnement de développement et d'exécution. Ecriture et exécution d'un premier programme.

### 3 Les variables

- Qu'est-ce qu'une variable ?
- Déclaration d'une variable var et let.
- Convention de nommage.
- Affectation / modification de valeur.
- Utilisation de constantes.
- Affichage par concaténation.

#### Travaux pratiques

Ecriture de plusieurs programmes simples manipulant les variables.

### 4 Type de données et opérateurs

- Qu'est-ce qu'un type ?
- Types primitifs : boolean, number, string, null, undefined.
- Les opérateurs d'affectation, de comparaison, arithmétiques et logiques.
- Précérence et l'associativité des opérateurs.

#### Travaux pratiques

Écriture de plusieurs programmes manipulant les types et les opérateurs.

## 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.

## 5 Structures de contrôle conditionnelle

- Les alternatives ou instructions conditionnelles if, else if, else.
- Opérateurs de comparaison pour structure conditionnelle.
- Les instructions conditionnelles : les ternaires.
- Les instructions conditionnelles : l'instruction switch.

### Travaux pratiques

Écriture de plusieurs programmes à structure conditionnelle.

## 6 Structure de contrôle itérative

- Les instructions itératives : boucles for.
- Les instructions itératives : boucles while.

### Travaux pratiques

Écriture de plusieurs programmes à structure itérative. Utilisation des structures de contrôle pour implémenter un algorithme.

## 7 Méthodes et fonctions

- Qu'est-ce qu'une méthode ? Qu'est-ce qu'une fonction ?
- Méthodes disponibles sur les chaînes de caractères et les nombres.
- Les expressions de fonction et leur déclaration.
- Paramètres, arguments et paramètres par défaut.
- Valeur de retour des fonctions.
- L'appel de fonction.
- Les fonctions de rappel (callback).
- Chaîne de portée et environnement lexical.

### Travaux pratiques

Écriture de plusieurs programmes avec manipulation de méthodes existantes. Création de fonctions avec et sans paramètre, utilisation des fonctions créées.

## 8 Les tableaux de données Array

- Qu'est-ce qu'un tableau ?
- Déclaration et affectation de tableau.
- Accéder aux propriétés d'un tableau.
- Modifier un élément dans un tableau.
- Itérer sur les éléments d'un tableau.
- Imbrication de tableaux.
- Introduction à la programmation fonctionnelle.

### Travaux pratiques

Écriture de plusieurs programmes avec création et manipulation de tableaux.

## 9 Introduction à la programmation Objet

- Qu'est-ce qu'un objet ?
- Déclaration d'objets.
- Les propriétés des objets.
- Itérer sur des objets.
- Le format JSON (JavaScript Object Notation).

### Travaux pratiques

Illustration des concepts Objets.

## 10 Maintenance, débogage

- Savoir lire et interpréter les différents messages d'erreurs.
- Utiliser un débogueur : exécuter un programme pas à pas, points d'arrêts, inspecter les variables pendant l'exécution.

### Travaux pratiques

Utilisation d'un débogueur pour contrôler l'exécution des programmes.

### Dates et lieux

#### CLASSE À DISTANCE

2026 : 25 mars, 27 mai, 5 oct.