



Vos compétences. Votre différence.

Référentiel pédagogique

JavaScript

Table des matières

Introduction au référentiel pédagogique	3
Le Tosa®	4
Objet du référentiel pédagogique	4
Une échelle de score unique	4
Domaines et sous-domaines de compétences.....	6
Niveau 1 - Initial.....	7
Synthèse	8
Niveau 2 - Basique	9
Langage et syntaxe.....	10
Structures de données et objets	10
Packages et API	11
JavaScript appliqué.....	12
Synthèse	12
Niveau 3 - Opérationnel.....	14
Langage et syntaxe.....	15
Structures de données et objets	15
Packages et API	15
JavaScript appliqué.....	16
Synthèse	16
Niveau 4 - Avancé	18
Langage et syntaxe.....	19
Structures de données et objets	19
Packages et API	19
JavaScript appliqué.....	20
Synthèse	21
Niveau 5 - Expert.....	22
Langage et syntaxe.....	23
Structures de données et objets	23
Packages et API	24
JavaScript appliqué.....	24
Synthèse	25

Introduction au référentiel pédagogique

Pour l'évaluation et la certification Tosa®

Le Tosa®

Les tests d'évaluation et les certifications Tosa® permettent de déterminer le niveau d'un candidat en évaluant ses compétences et ses aptitudes sur les logiciels bureautiques, les outils digitaux et les langages de programmation utilisés dans un environnement professionnel.

Ces tests sont ainsi conçus pour valider les compétences professionnelles des candidats souhaitant améliorer leur employabilité (salariés, étudiants, demandeurs d'emploi, personnes en reconversion).

Les évaluations et certifications Tosa® sont des tests adaptatifs, élaborés selon des méthodologies scientifiques (la détermination du score est basée sur l'Item Response Theory (IRT)). L'algorithme des tests s'adapte à chaque réponse du candidat pour ajuster le niveau de difficulté des questions qui lui sont posées, jusqu'à parvenir à la définition exacte de son niveau en calculant la limite haute de ses compétences. Les tests délivrent ainsi un diagnostic détaillé et unique sur les compétences de chaque candidat.

La robustesse et la fiabilité des tests Tosa® tiennent donc à l'association d'un modèle mathématique d'analyse de la difficulté des questions et à la pertinence des questions posées à chaque candidat (IRT).

C'est par ailleurs un modèle très proche de celui utilisé par le GMAT.

Objet du référentiel pédagogique

Ce référentiel pédagogique présente l'ensemble des compétences évaluées dans les domaines et sous-domaines des tests d'évaluation et de certification Tosa® JavaScript.

Les solutions d'évaluation et de certification Tosa permettent de situer le niveau de maîtrise des apprenants sur une échelle de score unique, allant de 0 à 1000 pour la certification, et divisée en cinq niveaux, d'« Initial » à « Expert », pour l'évaluation.

L'objet de ce référentiel est donc de préciser les connaissances techniques attendues sur chaque niveau, et dans chacune des quatre grandes catégories de compétences du langage de programmation JavaScript. Il doit ainsi permettre de déterminer les programmes d'enseignement ou de formation les plus adaptés à l'objectif de score d'un apprenant.

Une échelle de score unique

L'évaluation et la certification Tosa® reposent sur une échelle de score unique, traduite en cinq niveaux :

- d'Initial à Expert, pour l'évaluation ;
- de 1 à 1000 pour la certification.

Niveaux Tosa®	Scores Tosa®
Expert	876 - 1000
Avancé	726 – 875
Opérationnel	551 – 725
Basique	351 – 550
Initial	1 – 350

La certification Tosa JavaScript est délivrée avec indication d'un score (entre 551 et 1000), correspondant à un niveau (Opérationnel, Avancé ou Expert). En deçà du score de 551 points le candidat se verra délivrer une attestation de passage de la certification.

L'évaluation Tosa JavaScript est quant à elle délivrée avec indication d'un niveau allant d'Initial à Expert.

Domaines et sous-domaines de compétences

Langage et syntaxe	<ul style="list-style-type: none"> ■ Connaître les principales caractéristiques et spécificités de la langue ■ Connaître les fonctionnalités de base et la syntaxe ■ Gérer les entrées, les sorties et les erreurs d'un programme
Structures de données et objets	<ul style="list-style-type: none"> ■ Créer et manipuler des objets primitifs, des objets intégrés ou des objets personnalisés plus complexes ■ Comprendre et manipuler les propriétés d'un objet ■ Comprendre et manipuler le prototype des objets
Packages et API	<ul style="list-style-type: none"> ■ Créer des fonctionnalités partageables et réutilisables ■ Récupérer les données d'un serveur ■ Créer et manipuler des fonctions asynchrones ■ Gérer les cas d'erreurs et de succès lors de l'exécution d'un programme
JavaScript appliqué	<ul style="list-style-type: none"> ■ Ecrire un programme manipulant différents types de données de façon optimisée ■ Comprendre et trouver des erreurs à l'intérieur d'un code et les corriger ■ Rechercher des informations dans la documentation ■ Lire et comprendre une stack-trace afin de déduire l'origine d'un bug

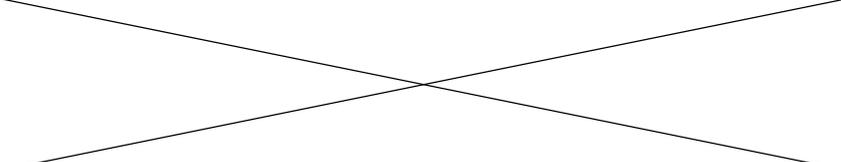
Niveau 1 - Initial

Entre 1 et 350 points

Le niveau initial pour un test d'évaluation est le niveau le plus bas sur l'échelle de score Tosa®. Il correspond au niveau d'un candidat qui n'a que très peu utilisé JavaScript ou qui n'a des notions que très parcellaires et limitées du fonctionnement du langage.

L'obtention du niveau initial signifie que le candidat connaît peu, voire pas du tout, les fonctionnalités de base de JavaScript, et qu'il ne peut l'utiliser dans un environnement professionnel.

Synthèse

Domaines	Compétences
Langage et syntaxe	<ul style="list-style-type: none">  Créer et utiliser une variable  Commenter une ligne de code  Utiliser des opérateurs arithmétiques basiques  Afficher le contenu d'une variable dans la console
Structures de données et objets	<ul style="list-style-type: none">  Reconnaître et manipuler des objets globaux  Reconnaître, créer, affecter et lire le contenu d'objets  Reconnaître, créer, affecter et lire le contenu de tableaux
Packages et API	<ul style="list-style-type: none">  Trouver un élément dans le DOM
JavaScript appliqué	

Niveau 2 - Basique

Entre 351 et 550 points

Préalablement à l'acquisition des compétences du niveau Basique, le candidat aura maîtrisé les compétences du niveau Initial.

Langage et syntaxe

- Maîtriser les outils d'entrée-sortie, dans le but de lire et modifier des documents externes au programme.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent d'automatiser le traitement de données à partir de documents externes, et sauvegarder ce traitement pour un archivage ou une réutilisation future.

- Maîtriser l'usage des commentaires en JavaScript dans le but de commenter son code.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent de commenter les morceaux de code complexes afin d'en rendre la compréhension plus facile pour les autres membres de l'équipe.

- Utiliser les fonctionnalités de contrôles de flux afin d'automatiser un traitement de données conditionnel et/ou répétitif.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent de reconnaître les cas d'usage typique de JavaScript et de mettre en place des processus d'extraction de données à partir d'une source connue et stable.

- Créer des fonctions simples et réutilisables.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent de créer des fonctions performant des tâches simples afin de résoudre plus rapidement un même problème se présentant plusieurs fois.

Structures de données et objets

- Comprendre et utiliser les types null et undefined afin de matérialiser l'absence de données.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent d'identifier l'absence de données (vérification du remplissage des champs d'un formulaire).

- Connaître et utiliser les fonctions basiques associées aux chaînes de caractères et aux tableaux.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent de ne pas avoir à redévelopper systématiquement le même lot de fonctions d'un programme à l'autre et ainsi gagner en temps et en qualité.

- Copier des objets partiellement ou les modifier partiellement sans avoir à recréer l'objet.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent de créer des programmes dont les données propres sont modifiables : par exemple, permettre à des utilisateurs de modifier leurs données personnelles.

Packages et API

- Maîtriser les concepts liés aux modules en vue d'inclure des fonctionnalités d'un module à un programme.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent de créer un programme réutilisant des fonctions et objets élémentaires créés par un autre membre de l'entreprise.

- Utiliser des fonctions provenant d'API telles que Math ou JSON.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent de créer un programme effectuant des opérations mathématiques avancées et pouvant produire des fichiers de sortie structurés.

- Pouvoir réagir aux événements du DOM.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent de créer un site internet avec lequel un utilisateur peut interagir de manière plus personnalisée.

JavaScript appliqué

- 🔧 Produire du code respectant des spécifications données.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent d'implémenter une spécification technique formulée par un manager.

- 🔧 Rechercher l'origine d'un problème simple dans un programme et le corriger.

Application métier : Par exemple, pour un profil de Développeur JavaScript, ces compétences permettent de corriger un problème existant dans un programme donné.

Synthèse

Domaines	Compétences
Langage et syntaxe	<ul style="list-style-type: none"> 🔧 Utiliser les déclarations const et let 🔧 Créer des commentaires multilignes 🔧 Utiliser des opérateurs arithmétiques booléens 🔧 Créer un workflow avec décisions conditionnelles 🔧 Créer des boucles simples 🔧 Créer sa propre fonction 🔧 Lire une entrée utilisateur à partir de la console
Structures de données et objets	<ul style="list-style-type: none"> 🔧 Reconnaître les types undefined et null 🔧 Appliquer une conversion de type sur une primitive 🔧 Utiliser des fonctions basiques sur les strings 🔧 Créer des propriétés complexes et les affecter 🔧 Utiliser des fonctions basiques sur les tableaux
Packages et API	<ul style="list-style-type: none"> 🔧 Importer un module à partir d'un fichier 🔧 Utiliser des fonctions basiques de l'API Math 🔧 Reconnaître et rédiger un document JSON 🔧 Gérer les événements DOM basiques

JavaScript appliqué	<ul style="list-style-type: none"><li data-bbox="564 297 1358 331">❏ Parcourir un tableau et en renvoyer une version modifiée<li data-bbox="564 356 1378 389">❏ Détecter une erreur conditionnelle dans une boucle simple
------------------------	--

Niveau 3 - Opérationnel

Entre 551 et 725 points

Préalablement à l'acquisition des compétences du niveau Opérationnel, le candidat aura maîtrisé les compétences du niveau Basique.

Langage et syntaxe

- ❖ Rédiger de la documentation technique de code (JSDoc).

Application métier : Par exemple, pour un profil de développeur JavaScript, ces compétences lui permettent de créer une bibliothèque de fonctions accompagnée d'une documentation technique générée sur la base des commentaires JSDoc.

- ❖ Maîtriser les outils d'entrée-sortie, dans le but de lire et modifier des documents externes au programme.

Application métier : Par exemple, pour un profil de développeur JavaScript, ces compétences lui permettent d'automatiser le traitement de données à partir de documents externes, et de sauvegarder ce traitement pour un archivage ou une réutilisation future.

- ❖ Maîtriser les fonctions fléchées afin de privilégier une syntaxe plus concise en comparaison avec les fonctions traditionnelles.

Application métier : Par exemple, pour un profil de développeur JavaScript, ces compétences lui permettent de créer du code plus succinct, et de maintenir facilement une base de code lisible.

Structures de données et objets

- ❖ Créer n'importe quel type de chaînes de caractères, quel que soit leur contenu.

Application métier : Par exemple, pour un profil d'analyste, ces compétences lui permettent d'afficher tout type de messages sans plantage du programme.

- ❖ Identifier le type d'une variable lors de l'exécution et la traiter de manière appropriée.

Application métier : Par exemple, pour un profil d'analyste, ces compétences lui permettent de gérer des données sous différentes formes et de les traiter selon leurs spécificités.

Packages et API

- ❖ Créer un module simple pour gérer des fonctionnalités partageables et réutilisables.

Application métier : Par exemple, pour un profil de développeur Web, ces compétences lui permettent de regrouper des fonctions et objets préalablement créés dans un module en vue d'une réutilisation cohérente.

- Maîtriser l'asynchronicité JavaScript.

Application métier : Par exemple, pour un profil de développeur Web, ces compétences lui permettent d'exécuter des programmes réactifs, attendant des informations extérieures sans bloquer l'affichage et les interactions de la page.

- Faire des appels HTTP à des serveurs distants.

Application métier : Par exemple, pour un profil de développeur Web, ces compétences lui permettent de créer un client web léger se connectant à un serveur distant afin de piloter et récupérer des données externes.

JavaScript appliqué

- Rechercher de la documentation en ligne pour les besoins de son programme.

Application métier : Par exemple, pour un profil de développeur Web, ces compétences lui permettent de se former à l'utilisation de technologies basiques d'un projet existant.

Synthèse

Domaines	Compétences
Langage et syntaxe	<ul style="list-style-type: none"> Utiliser un opérateur d'affectation Créer de la documentation standard JSDoc Gérer une comparaison de type égalité stricte Configurer des décisions conditionnelles complexes dans un workflow Maîtriser les instructions de contrôle de flux associées aux boucles Créer des fonctions fléchées Gérer les propriétés dont les valeurs sont indéfinies
Structures de données et objets	<ul style="list-style-type: none"> Identifier les types associés aux variables à l'exécution Echapper n'importe quel caractère au sein d'une chaîne de caractères Créer une nouvelle référence objet Effectuer des opérations complexes sur les tableaux

Packages et API	<ul style="list-style-type: none"> 🔗 Créer un module et exporter des fonctions 🔗 Sérialiser et désérialiser JSON 🔗 Comprendre les mécanismes asynchrones 🔗 Gérer les cas de réussite et d'échec d'une opération asynchrone 🔗 Ajouter un élément dans le DOM 🔗 Récupérer les données d'un serveur
JavaScript appliqué	<ul style="list-style-type: none"> 🔗 Parcourir un tableau selon une caractéristique donnée 🔗 Trouver une erreur simple dans une portion de code donnée 🔗 Trouver la documentation associée à une librairie JavaScript simple

Niveau 4 - Avancé

Entre 726 et 875 points

Préalablement à l'acquisition des compétences du niveau Avancé, le candidat aura maîtrisé les compétences du niveau Opérationnel.

Langage et syntaxe

- Créer un programme pour des traitements complets de données, en utilisant les fonctionnalités avancées des contrôles de flux et des opérations arithmétiques.

Application métier : Par exemple, pour un profil de Webmaster, ces compétences permettent d'agréger des données disponibles à partir d'une extraction existante.

- Maîtriser les comparaisons abstraites et strictes des types primitifs.

Application métier : Par exemple, pour un profil de Webmaster, ces compétences permettent d'écrire des comparaisons avec un type de comparaison adapté, ainsi que de réduire les bugs liés aux comparaisons de variables.

Structures de données et objets

- Maîtriser les littéraux de gabarits et générer des chaînes de caractères suivant un format préétabli de manière simple et lisible.

Application métier : Par exemple, pour un profil de Webmaster, ces compétences permettent d'afficher des messages dont le contenu diffère selon l'état du programme.

- Choisir les structures de données les plus appropriées aux besoins.

Application métier : Par exemple, pour un profil de Webmaster, ces compétences permettent d'améliorer les performances d'un programme.

Packages et API

- Renommer les imports de modules.

Application métier : Par exemple, pour un profil de Webmaster, ces compétences permettent d'utiliser des modules externes ayant les mêmes noms de fonctions sans apparition de conflits.

- Maîtriser la manipulation du DOM (ajout, retrait, remplacement d'éléments).

Application métier : Par exemple, pour un profil de Webmaster, ces compétences permettent de créer une page web dynamique dont le contenu évolue selon des événements externes ou internes. (Ex : Création d'une page de messagerie instantanée)

JavaScript appliqué

- 📌 Lire et comprendre une stack-trace afin de déduire l'origine d'un bug.

Application métier : Par exemple, pour un profil de Webmaster, ces compétences permettent de reproduire du bug puis d'analyser le message d'erreur dans le but de déduire l'origine du bug et le résoudre.

Synthèse

Domaines	Compétences
Langage et syntaxe	<ul style="list-style-type: none">  Gérer une comparaison de type égalité abstraite  Maîtriser les opérations arithmétiques pour différents types  Maîtriser la configuration des décisions conditionnelles dans un workflow
Structures de données et objets	<ul style="list-style-type: none">  Créer des littéraux de gabarits  Manipuler les propriétés d'un objet  Créer un tableau à partir de plusieurs autres tableaux  Identifier les collections
Packages et API	<ul style="list-style-type: none">  Renommer les modules importés  Créer un regex qui correspond à une chaîne de caractères  Remplacer des éléments du DOM  Gérer plusieurs opérations asynchrones
JavaScript appliqué	<ul style="list-style-type: none">  Ecrire du code capable de lire les données du DOM et de les traiter  Trouver l'origine d'une erreur en utilisant un navigateur et en connaissant uniquement la trace de la pile de l'exception  Trouver et comprendre une documentation afin d'apprécier les sorties standard et d'erreur d'une portion de code existante

Niveau 5 - Expert

Entre 876 et 1000 points

Préalablement à l'acquisition des compétences du niveau Expert, le candidat aura maîtrisé les compétences du niveau Avancé.

Langage et syntaxe

- Effectuer une opération bit à bit avec l'opérateur adéquat.

Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent d'effectuer des conversions et calculs numériques afin d'exploiter la rapidité des opérations bit à bit par rapport à leurs équivalents traditionnels Math et parseInt.

- Connaître la précision des nombres à virgule flottante.

Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent de détecter des opérations numériques impliquant des problèmes de précisions et de corrigé le résultat erroné.

- Maîtriser le changement de contexte d'une fonction.

Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent d'exécuter conformément une fonction lors de l'écriture d'un programme (Ex : Utilisation multiple d'une fonction de "callback").

- Détecter les erreurs lors de l'exécution d'un programme et les traiter.

Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent de gérer des cas d'erreurs provenant d'une anomalie lors de la réception des données du serveur ou de l'exécution du programme.

Structures de données et objets

- Comprendre les principes de prototype et leur utilisation.

Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent de créer des objets qui héritent de caractéristiques communes.

- Utiliser les différentes fonctions natives de manipulation de tableaux.

Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent de transformer des données stockées dans un tableau de façon optimisée en un objet simple prêt à être affiché.

Packages et API

- Créer un programme détectant une sous-expression à l'issue de l'exécution d'une regex.

Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent de détecter et renvoyer des balises html contenues dans une chaîne de caractères en vue d'une interprétation spécifique.

- Créer, écouter et réagir à des événements personnalisés venant du DOM (Document Object Model).

Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent d'écouter des champs de formulaire et de renvoyer une erreur pour des conditions de validation des champs non remplies.

- Créer des fonctions asynchrones dont la donnée de retour diffère en cas de succès et en cas d'erreur.

Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent de récupérer des données d'un serveur dans le but de renvoyer les données reçues en cas de succès ou l'erreur si détection d'une anomalie lors du traitement de la requête.

JavaScript appliqué

- Écrire un programme avec une logique avancée de manipulation des données.

Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent de trier les données de façon optimisée.

- Trouver et comprendre des erreurs à l'intérieur d'un code complexe. Les corriger.

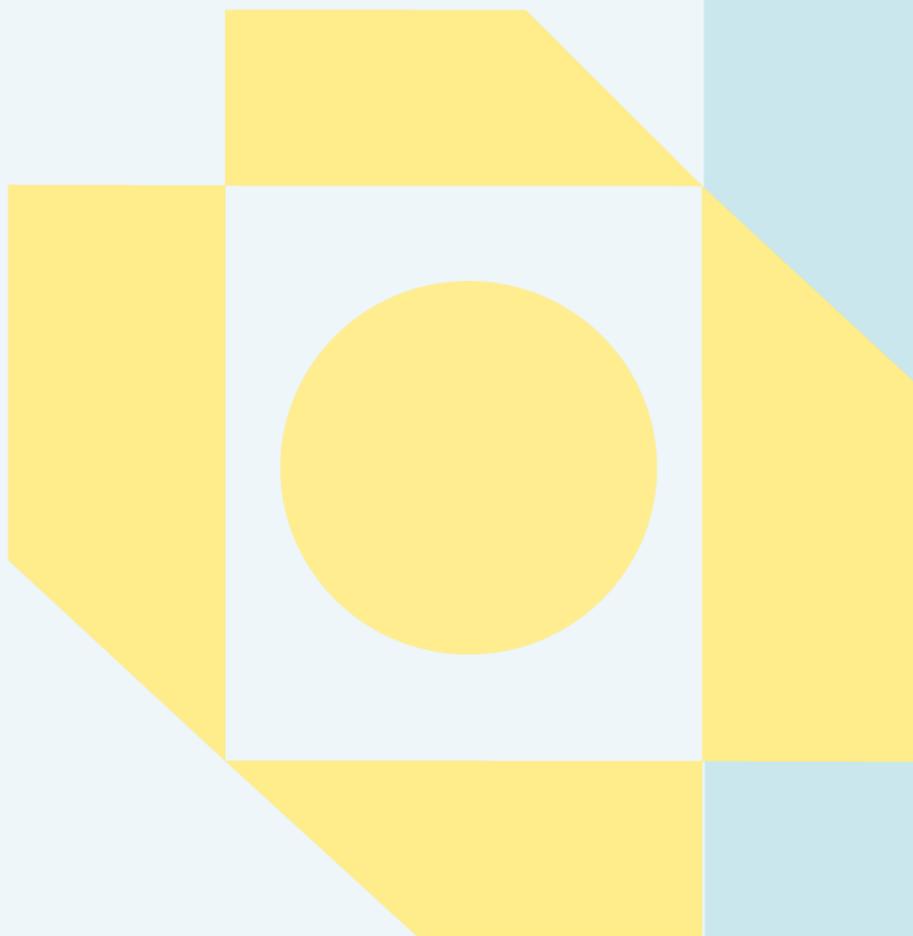
Application métier : Par exemple, pour un profil de développeur full-stack, ces compétences lui permettent de corriger une erreur et/ou contribuer à un code écrit par un membre d'équipe.

Synthèse

Domaines	Compétences
Langage et syntaxe	<ul style="list-style-type: none">Appliquer des opérations bit-à-bitPrendre en compte l'imprécision liée à un nombre à virgule flottanteMaîtriser la liaison de fonctionCapter et gérer les erreurs
Structures de données et objets	<ul style="list-style-type: none">Comprendre le principe des langages orientés prototypeMaîtriser les opérations s'appliquant aux tableaux
Packages et API	<ul style="list-style-type: none">Capter les sous-expressions lors de correspondances regexRéagir à des événements DOM complexesCréer des fonctions asynchrones
JavaScript appliqué	<ul style="list-style-type: none">Implémenter une logique complexe au sein d'une fonctionTrouver plusieurs erreurs au sein d'une portion de code complexe



Vos compétences. Votre différence.



contact@isograd.com
www.tosa.org