

+ -

Programmation Orientée Objet JAVASCRIPT

Consiste à utiliser du code autonome (objets) pour développer des applications. En JavaScript, cela peut être réalisé en créant un plan (class) pour fabriquer des objets.

Une **classe** encapsule un ensemble de propriétés (**fonction constructeur**) et de comportements (**fonctions de classe**) qui peuvent être utilisés pour instancier un **objet** de valeurs spécifiques. Ceci est généralement utilisé pour modéliser des objets du monde réel.

Vidéo 1

Introduction

C'est un paradigme de programmation centré autour des objets et non des fonctions.

La POO n'est ni un langage de programmation ni un outil de programmation. Plusieurs langage applique la POO comme: JAVA, PYTHON, C++, DART, PHP, RUBY

- Les 04 piliers de la POO:

ENCAPSULATION

C'est la protection de l'information contenu dans un objet en les rendant manipulable uniquement par certaines méthodes.

ABSTRACTION:

C'est se baser sur l'essentiel et non sur les détails.

HERITAGE:

C'est l'ensemble des attributs et méthodes transmis par une classe mère à une classe fille.

POLYMORPHISME

C'est la capacité d'un objet à pouvoir prendre plusieurs formes.

- Les avantages de la POO:

L'encapsulation réduit la complexité du code et permet de réutiliser le code .

L'abstraction réduit également la complexité du code pour qu'il soit plus maintenable.

L'héritage élimine la redondance du code.

Le polymorphisme permet de rétracter les mauvaises déclarations des Switch/ Case.

Vidéo 2

Value Types - Reference Types

- Value types encore appelé primitive type
- les différents types sont: number, boolean, string, null: appelé Values Types.
- référence types, c'est lorsqu'on a des objets et des tableaux

Vidéo 3

Programmation procédurale

La Programmation procédurale: Consiste à utiliser du code selon une procédure étape par étape pour développer des applications .

Vidéo 4

L'encapsulation

Object Literal: Le littéral objet est un type de données utilisé pour définir des objets en javascript. Il s'agit d'une syntaxe permettant de

créer un objet en javascript composé des paires clé-valeur. C'est un moyen efficace et léger de créer et de stocker des données.

Utilité: - Les littéral objet sont facile à lire et à comprendre
- Ils peuvent stocker tout type de donné

Vidéo 5

Factory Function

Factory Function ou Fonction d'usine, c'est une fonction (objet) qui renvoie un nouvel objet.

Vidéo 6

Constructor Function

Une fonction constructeur est une fonction régulière avec les conventions suivantes: Le nom commence toujours par une lettre majuscule et on l'appelle toujours avec l'opérateur "new".

Vidéo 7

Abstraction

Une classe abstraite est une classe qui ne peut pas être directement instanciée. Une classe abstraite encapsule des attributs et méthodes qui peuvent être utilisés par les instances des classes qui en héritent. L'intérêt des classes abstraites est de regrouper plusieurs classes sous un même nom de classe (par polymorphisme) ainsi que de décrire partiellement des attributs et méthodes communs à plusieurs classes.

Vidéo 8

Classe

Les classes sont des modèles qui permettent de créer des objets sur ce même modèle. La déclaration d'une classe regroupe des membres, méthodes et propriétés (attributs) communs à un ensemble d'objets. La différence entre une structure et une classe est que les attributs et méthodes d'une structure sont par défaut publics, alors qu'ils sont par défaut privés dans une classe. Une autre différence est que les structures ne peuvent utiliser l'héritage.

Vidéo 9 Héritage

L'héritage de classe est un moyen pour une classe d'étendre une autre classe. Elle est aussi un mécanisme qui permet d'inclure les caractéristiques d'une classe mère à une classe fille.

Vidéo 10 Polymorphisme

C'est la capacité d'un objet à prendre plusieurs formes.

Il existe 3 types de polymorphisme:

- Le **polymorphisme ad hoc** (également surcharge ou en anglais overloading): Le polymorphisme ad hoc permet d'avoir des fonctions de même nom, avec des fonctionnalités similaires, dans des classes sans aucun rapport entre elles (si ce n'est bien sûr d'être des filles de la classe objet).
- Le **polymorphisme** paramétrique (également généricité ou en anglais template): Le polymorphisme paramétrique, appelé généricité, représente la possibilité de définir plusieurs fonctions de même nom mais possédant des paramètres différents (en nombre et/ou en type).
- Le **polymorphisme** d'héritage (également redéfinition, spécialisation ou en anglais overriding): La possibilité de redéfinir une méthode dans des classes héritant d'une classe de base s'appelle la spécialisation.