JAVA
LES FONDAMENTAUX

Présenté par :

Xavier TABUTEAU

01. Introduction

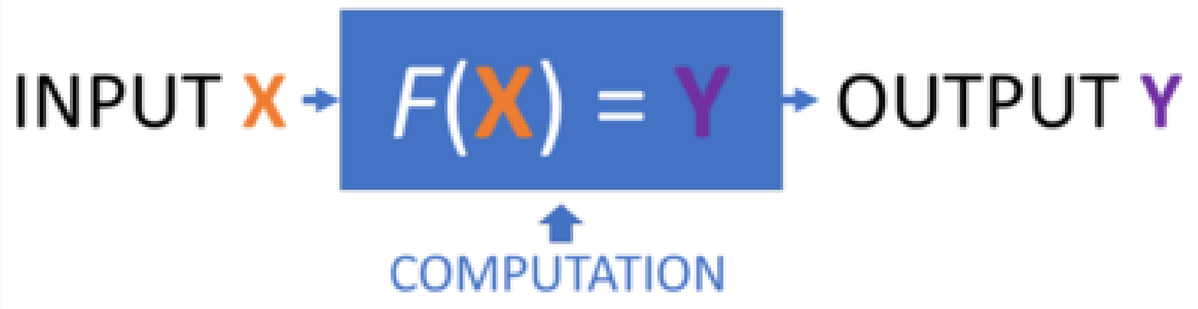
Développer ne consiste pas à écrire des formules mathématiques (sauf si vous travailler sur des logiciels destinées aux sciences évidemment). On parle plutôt **d'algorithme**.

Même si les langages sont différents, l'objectif principal d'un langage de programmation reste celui d'instruire la machine pour qu'elle produise des outputs conformes aux objectifs de l'application.

On résume souvent par :

F(X) = Y, où :

X représente l'input
Y représente l'ouput
F() représente la fonction qui permet de transformer X en Y



01. Introduction

La programmation permet de résoudre un problème de manière automatisée grâce à l'application d'un algorithme :

Programme = Algorithme + Données

Un algorithme est une suite d'instructions qui sont évaluées par le processeur sur lequel tourne le programme.

Les instructions utilisées dans le programme représente le **code source**.

Pour que le programme puisse être exécuté, il faut utiliser un langage que la machine peut comprendre : un **langage de programmation**.

01. Introduction

Il existe des langages procéduraux

Exemple : le langage C

Il existe des langages fonctionnant à base d'objets

Exemple : le langage Java

Documentation officielle de Java : https://docs.oracle.com/en/java/javase/index.html

Java est un langage de haut niveau

Langage bas niveau vs langage de haut niveau

Un langage de bas niveau est un langage qui est considéré comme plus proche du langage machine (binaire) plutôt que du langage humain. Il est en général plus difficile à apprendre et à utiliser mais offre plus de possibilité d'interactions avec le hardware de la machine.

Un langage de haut niveau est le contraire, il se rapproche plus du langage humain et est par conséquent plus facile à appréhender. Cependant les interactions se voient limitées aux fonctionnalités que le langage met à disposition.

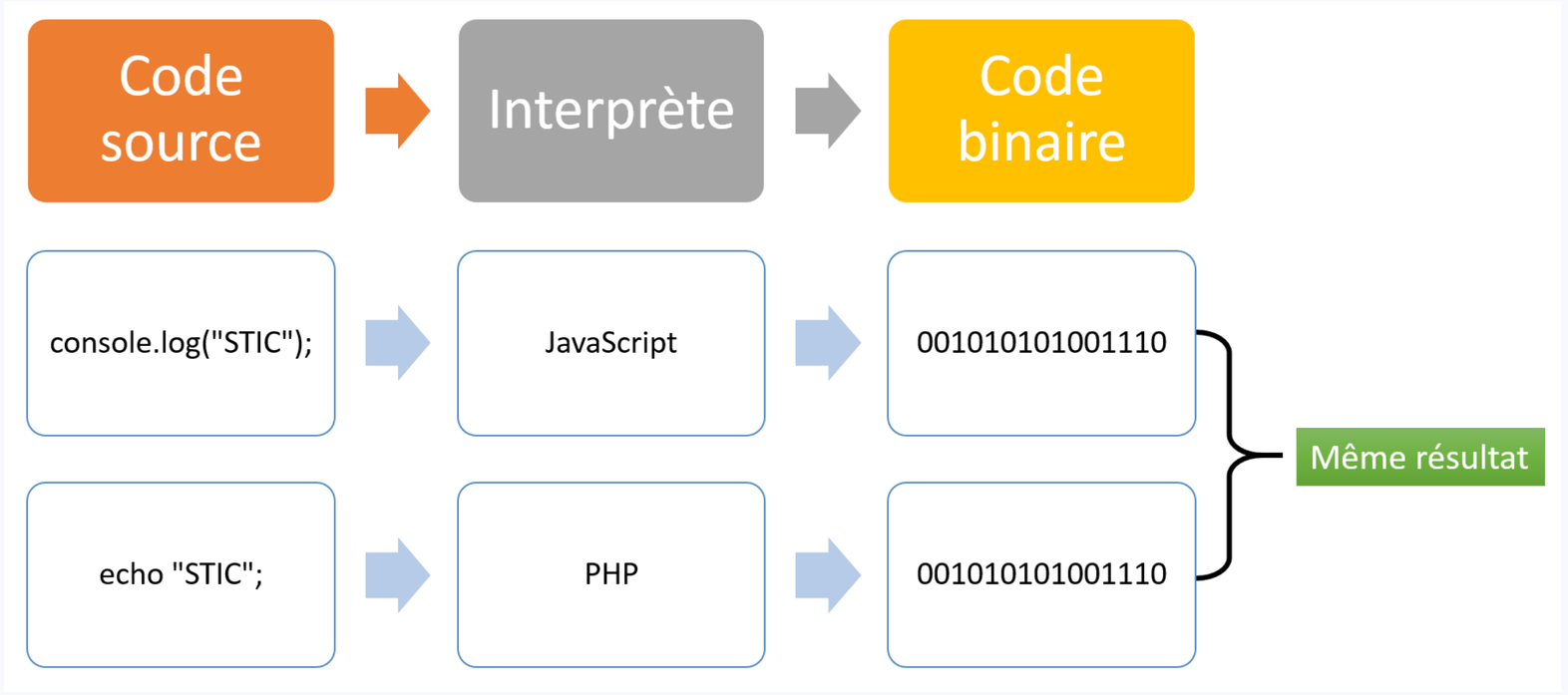
Java est un langage interprété

Compilation vs Interprétation

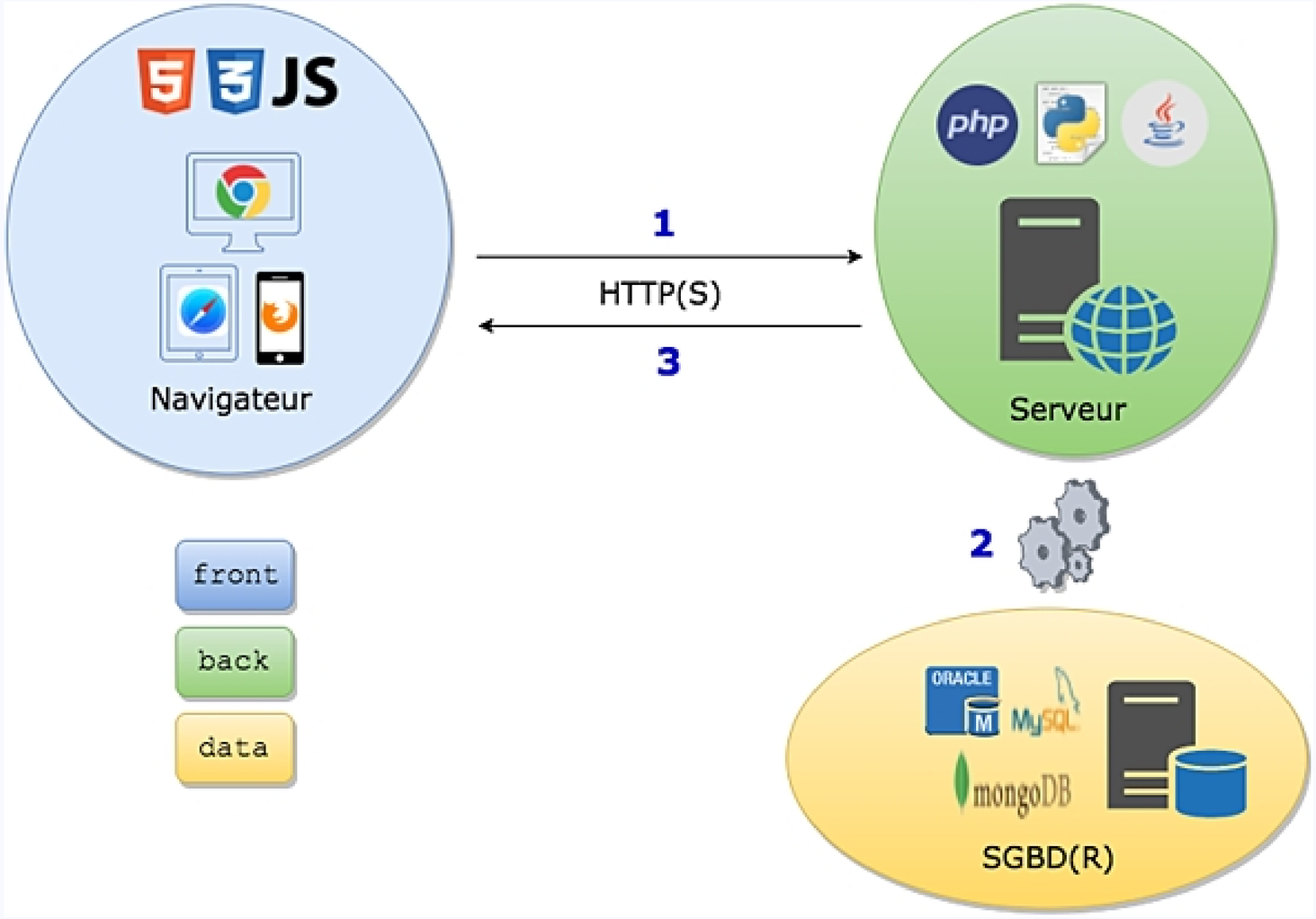
La compilation d'un programme consiste à transformer toutes les instructions en langage machine avant que le programme puisse être exécuté. Par conséquent il sera nécessaire de refaire la compilation après chaque modification du code source.

Si un langage n'est pas compilé, il est nécessaire d'utiliser un interprète qui traduit les instructions en temps réel (on run time). Dans ce cas le code source est lu à chaque exécution et par conséquent les changement apportés au code seront pris en compte directement. La contrainte réside dans le fait que le machine faisant tourner le programme doit disposer de l'interprète de celui-ci.

01. Introduction



Anatomie d’une application interactive



Architecture de Java

Le langage Java

- Ecriture des programmes Java
- Orienté objet
- Compilation du code Java en byte code (langage machine portable en fichier .javac)

Le Machine Virtuelle Java

- Interprète et exécute le byte code
- La machine virtuelle Java (JVM) est disponible pour chaque système d'exploitation
- Un fichier en byte code peut être exécuté sur n'importe quelle JVM indépendamment de l'OS

La plateforme Java

- Ensemble de classes prédéfinies
- API (Application Programming Interface) disponible pour les développeurs

Architecture de Java

Gestion de la mémoire

- L’interpréteur Java (JVM) sait quels emplacements mémoires il a alloué.
- Il sait déterminer quand un objet alloué n’est plus référencé par un autre objet ou une variable.
- Ramasse-miette (« Garbage Collector ») : détecte et détruit ces objets non référencés (libération automatique de la mémoire).

Java est un langage objet

- Tout est classe et objet !
- En Java, tout ce qui est produit est sous forme de classes.
- Les fonctionnalités de base de Java sont disponible sous forme de classes.
- C’est au développeur d’identifier ce qui doit être créé par rapport au problème à résoudre !

L’identification des classes est issue d’un processus d’analyse.

Il est possible d’utiliser une méthodologie pour identifier les classes.

Par exemple : UML (Unified Modeling language)

Java est un langage objet

Exemple :

- Un garage entretient des voitures
- Un garage est référencé par sont N° de Siret, nom, adresse, propriétaire, chiffre d'affaire et nombre de salariés
- Les voitures possèdent des caractéristiques comme la marque, un numéro de série
- Les voitures possèdent toutes un moteur et un châssis
- Le châssis a notamment un numéro de série
- Le moteur a notamment une référence et une puissance
- Les voitures roulent, démarrent, freinent

Java est un langage objet

Exemple :

Le diagramme de classes UML correspondant aurait :

- Une classe Garage, avec ses propriétés (adresse, etc...)
- Une classe Voiture, avec ses propriétés (marque, etc...)
- Une classe Chassis, avec ses propriétés (poids , etc...)
- Une classe Moteur, avec ses propriétés (puissance , etc...)
- Une relation entre Garage et Voiture (entretien)
- Une relation d'agrégation entre Voiture et Chassis et Moteur, Voiture étant l'agrégat (composée)

Java est un langage objet

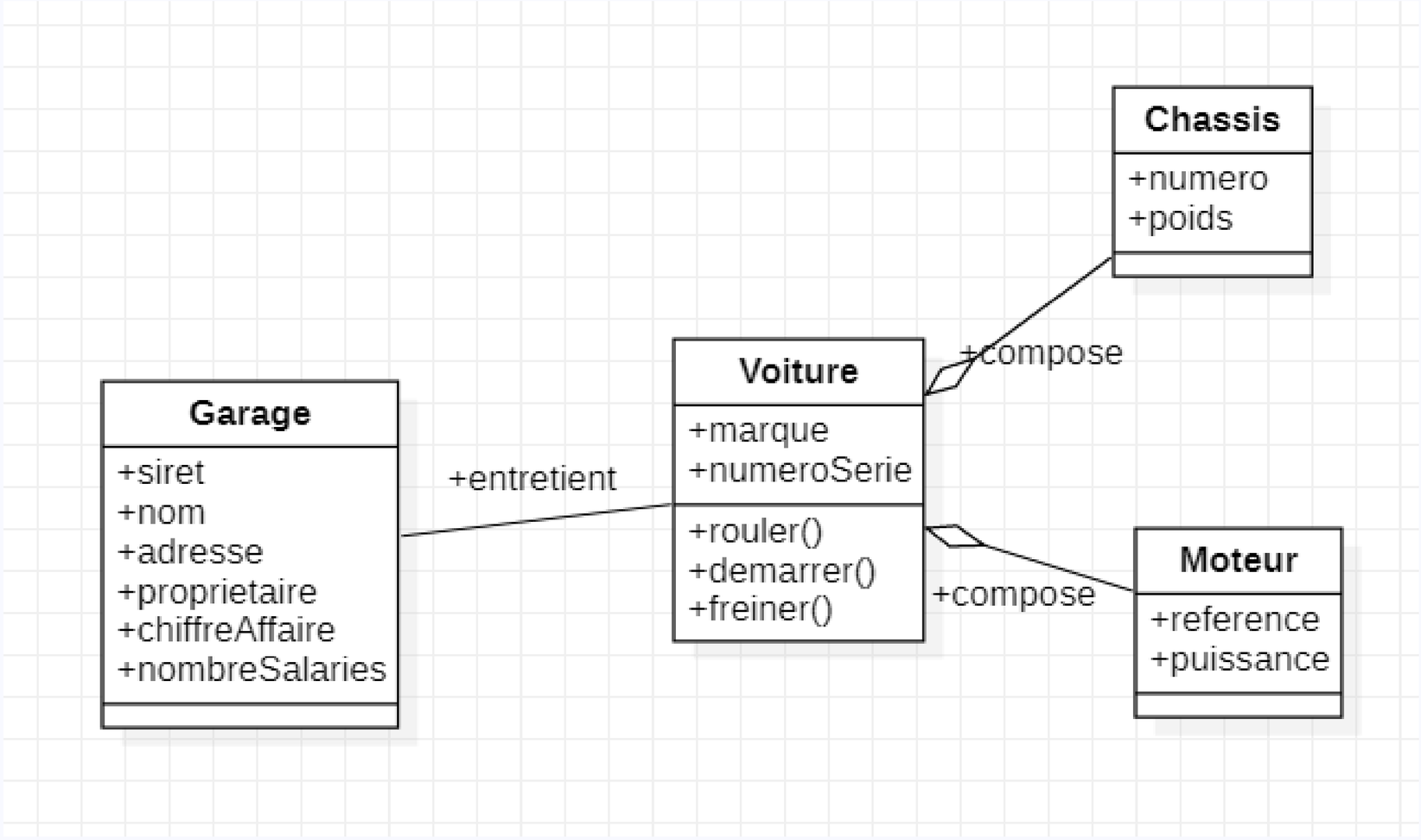
Exemple :

Faire le diagramme de classe avec StartUML : https://staruml.io/download

- Classe Garage avec ses attributs
- Classe Voiture, avec ses attributs et méthodes
- Classe Chassis, avec ses attributs
- Classe Moteur, avec ses attributs
- Relation entre Garage et Voiture (entretien)
- Relation d'agrégation entre Voiture et Chassis et entre Voiture et Moteur
  (Voiture comporte un châssis et un moteur)

Java est un langage objet

Exemple :



JAVA
LES FONDAMENTAUX

Présenté par :

Xavier TABUTEAU