

Chapitre 5

La programmation orientée objet

Il y a trois paradigmes (façons de programmer) de programmation largement utilisés : La programmation procédurale : le code est un enchaînement de procédures pour résoudre le problème. La programmation fonctionnelle : le code est un enchaînement de fonctions pour résoudre le problème. La programmation orientée objet (POO) : on identifie les acteurs (=objets) du problème puis on détermine la façon dont ils doivent interagir pour résoudre le problème.

La programmation orientée objet est une nouvelle façon de concevoir et de développer des applications informatiques. Elle permet une plus grande modularité, ainsi qu'une meilleure lisibilité du code et une meilleure maintenabilité du code.

5.1 Définitions d'objet et classe

La POO consiste à représenter des objets du monde réel dans un programme informatique.

Pour représenter un objet du monde réel dans un programme informatique, on considère un ensemble de données appelées aussi attributs ou propriétés de cet objet et un ensemble d'actions appelées aussi méthodes de cet objet.

Chaque attribut ou propriété est caractérisé par sa valeur. Une méthode s'appelle aussi fonction membre, est une action applicable à son objet.

Exemples d'objet :

- compte bancaire : a pour attributs ou propriétés numéro, nom de son propriétaire, solde, type de compte etc et comme méthodes débiter, accrediter, afficher solde, bloquer, etc
- footballeur : a pour attributs ou propriétés nom, numéro qu'il porte etc et a pour actions ou méthodes dribbler, courir, attraper le ballon, marquer le but etc
- voiture : a pour attributs marque, modèle, couleur, et a comme actions ou méthodes démarrer, rouler, accélérer etc...

Représentation graphique d'une classe. Voir Table 5.1

- Etudiant : nom, code, filière, modules suivis, notes etc et a comme actions ou méthodes valider, passer session de rattrapage, etc

Une classe est une description d'un ensemble d'objets ayant une structure de données commune (attributs ou propriétés) et disposant des mêmes méthodes (ou actions). Ainsi, tous les objets d'une même classe ont en commun les mêmes propriétés et les mêmes méthodes.

Une classe est caractérisée par ses attributs ou propriété et ses méthodes.

Classe : voiture
Attribus ou propriétés
Marque Modèle Couleur ...
Action ou méthodes
Démarrer rouler accélérer ...

TABLE 5.1 – Une classe voiture

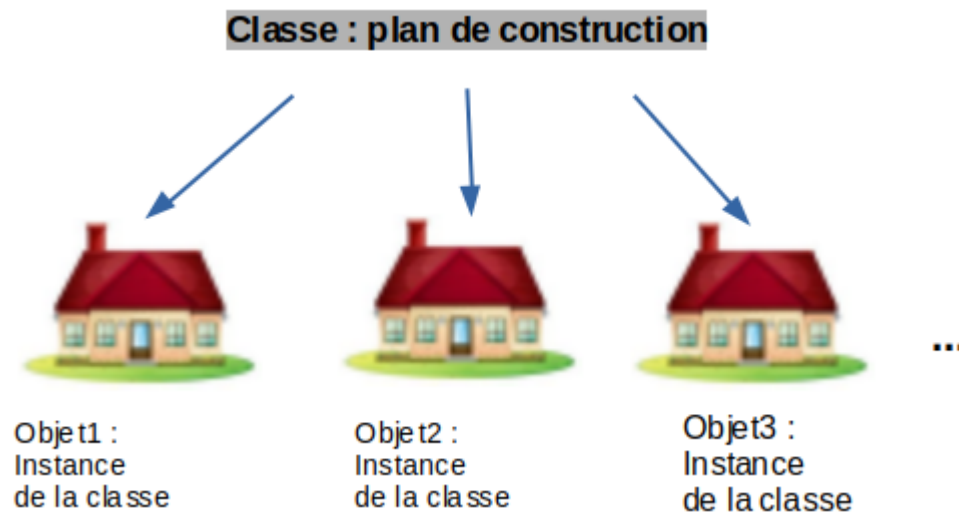


FIGURE 5.1 – Instanciation : création d’objets à partir d’une classe

Une classe peut être vue comme un modèle d’objet ou comme un moule (voir Figure 5.2) qui permet de créer autant d’objets de même type et de même structure. L’instanciation est l’opération de créer un objet d’une classe. La notion de classe généralise celle de type.

Dans la pratique, on commence par créer une classe en déclarant ses propriétés, et programmant ses méthodes. Lorsqu’on déclare qu’un objet est de cette classe, alors cet objet obtient automatiquement toutes les propriétés et méthode définis dans sa classe. Il ne reste plus qu’à spécifier les valeurs des propriétés pour chaque objet.

Certains langages de programmation supportent la POO et d’autres non. Par exemple C n’est pas orienté objet. Par contre son extension C++ l’est.

Il y a deux grandes catégories de langages à objets :

- les langages à classes, comme C++, Java, Python et d’autres.
- les langages à prototypes, comme JavaScript, NewtonScript, Self, Lua et d’autres.

La programmation orientée prototype est un style de programmation orientée objet qui n’utilise pas les classes.



FIGURE 5.2 – Le moule (la classe) permet de créer autant d’objets qu’on veut

5.2 JavaScript et programmation orienté objet

Comme JavaScript est spécialement fait pour améliorer le langage HTML, fait que la plus part de ses objets utilisés sont prédéfinis ainsi que leur propriétés et méthodes. La Figure 5.3 présente tous les objets prédéfinis en JavaScript . On peut exceptionnellement définir des classes et des objets en JavaScript , mais ceux prédéfinis sont largement suffisants.

Les mécanismes relatifs à la programmation orientée objet sont différents de ceux de Java ou C++. En JS la notion de classe et le concept d’héritage n’existent.

Les méthodes se terminent toujours avec des parenthèses. Elles peuvent recevoir ou non des paramètres. Les propriétés ne supportent pas les parenthèses.

En Javascript, pour accéder à une propriété d’un objet, on utilise la syntaxe :

```
Objet.propriété
```

et pour modifier une propriété

```
Objet.propriété=nouvelleValeurDePropriété
```

Par exemple

```
<html>
<body><head><title>Ma page Web </title></head></body>
<script type="text/javascript">
document.write(document.title); // retourne 'Ma page Web'
document.title='Ma nouvelle page Web'
// le titre de la fenêtre devient 'Ma nouvelle page Web'
</script>
</body>
</html>
```

Certaines propriétés sont modifiables, par contre d’autres sont en lecture seule, elles permettent uniquement de récupérer des informations sur les objets.

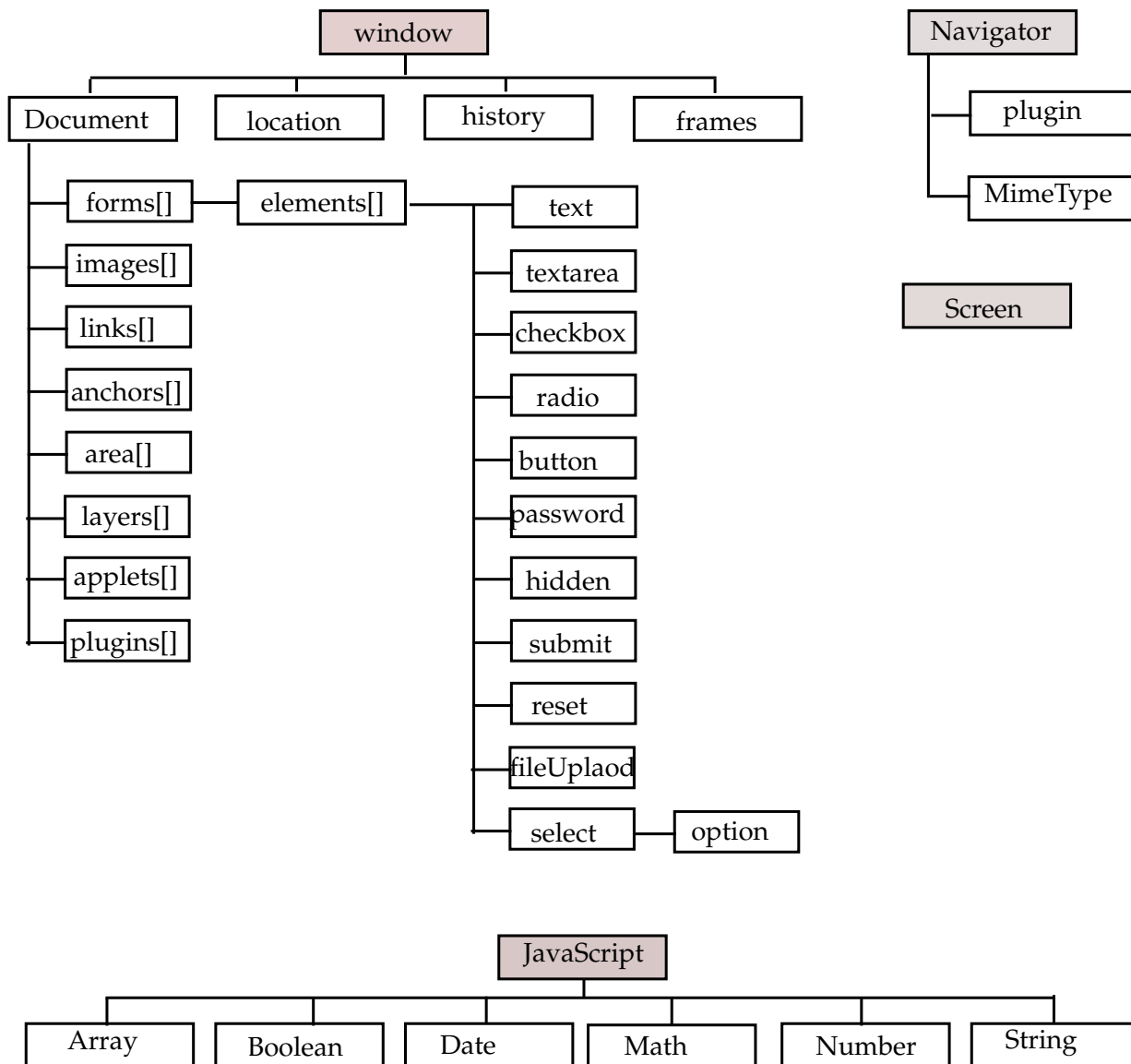


FIGURE 5.3 – Les objets prédéfinis en JavaScript

Pour définir un nouveau objet

```
var monObjet = new Object();
```

Par exemple

```
<script type="text/javascript">
var maDte=new Date();//instanciation de l'objet maDte
document.write(maDte);
// retourne Thu Jan 28 2021 14:23:24 GMT+0100 (UTC+01:00)
document.write(maDte.getFullYear());
// retourne 2021
maDte.setFullYear(2029);
document.write(maDte.getFullYear());
// retourne 2029
</script>
```

```
Objet.methode()
```

Par exemple :

alert() est une méthode de l'objet window, on écrit window.alert("text d'alerte").

write() est une méthode de l'objet document, on écrit document.write("text1","text2",...).

Dans le prochain chapitre, nous allons voir pour chaque objet ses propriétés et ses méthodes.