

## Langages et programmation

Cette fiche de révision appartient au chapitre « Langages et programmation ». Les notions suivantes sont abordées : les bonnes pratiques de programmation et les différents paradigmes de programmation .

Il n'existe pas de langage de programmation « universel ». En effet, si l'on peut, en général, **tout programmer avec n'importe quel langage**, certains langages seront plus adaptés à la programmation de telle ou telle tâche. Le langage Python, par exemple, est très utilisé pour la mise en place rapide de programmes. Cependant, dès lors que l'on a besoin de performances plus élevées, ce n'est plus forcément le langage le plus adapté.

Il en résulte que depuis la création de l'informatique de nombreux langages sont apparus, souvent inspirés de langages déjà existants et incorporant ou développant de nouveaux aspects. Le temps d'apprendre à coder dans un langage, celui-ci est déjà dépassé et remplacé par un nouveau langage.

### Les bonnes pratiques de programmation:

Avant de se lancer dans la programmation, l'expression claire d'un énoncé est indispensable pour tenir compte de toutes les particularités du problème (besoin fonctionnel).

On **spécifie** son code, son programme , sa fonction en donnant :

- L'auteur et le nom du programme.
- La description des données en entrée (quelles types de variables).
- Une description de ce que doit faire le programme.
- La description des données en sortie.

En conséquence peut importe le langage, un code "**bien écrit**" doit avoir les propriétés suivantes :

- Être facile à lire, pas soi-même mais aussi par les autres.
- Avoir une organisation logique et évidente, utiliser une charte de nommage.
- Être explicite, montrer clairement les intentions du développeur.
- Être soigné et spécifié.

Un tel code sera dès lors facilement « **supportable** » (le faire durer et évoluer dans le temps).

**Les commentaires** sont essentiels pour "éclairer" le code. Un commentaire est un texte qui est ignoré par l'ordinateur lorsqu'il exécute le programme, mais qui peut être lu par le développeur lorsqu'il lit le programme. Bien que les commentaires soient essentiels, il ne faut pas en abuser.

Il est important d'utiliser des **outils de versionning** comme **GitHub, SVN, ...** pour

- Partager le code
- Mettre en commun des modules
- Gérer l'historique, les droits et workflows

Les méthodes **Agiles** sont des groupes de pratiques de pilotage et de réalisation de projets. La plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée .

## Les différents paradigmes de programmation:

En programmation, un **paradigme** est la manière d'écrire un programme basé sur un ensemble de principes ou une théorie.

**Programmation impérative** : Décrit les opérations en séquences d'instructions exécutées dans l'ordre (procédurale). C'est celui des premiers langages de programmation (C, Basic, Pascal, . . . )

**Paradigme fonctionnel** : Tout est fonction, absence de variables mutables, une fonction peut appeler une autre fonction ou s'appeler elle-même (Réursive), on utilise de préférence des fonctions pures. La plupart des langages (C, Python, ...) autorise le paradigme fonctionnel.

Une **fonction pure** est une fonction qui respecte ces deux propriétés:

- A toujours les mêmes valeurs de retours pour les mêmes entrées
- N'a pas d'effet de bord

**Programmation objet** : S'intéresse aux données plutôt qu'aux traitements.

Toute entité du monde est un objet défini par une classe. Un objet peut contenir des attributs ou *propriétés* ainsi que des actions ou *méthode*. Le langage C++ est une évolution du langage C pour permettre l'utilisation de la programmation objet. Le « diagramme de classe UML » est une méthode qui standardise la façon de structurer les objets.

**Programmation événementielle** : Le déroulement du programme est déterminé par des événements. Ce type de programmation est utilisé dans les automates, drones, droid, ou des langages comme Javascript qui interagissent avec un utilisateur ( ou un capteur pour les robots).

### CE QU'IL FAUT RETENIR

Il faut suivre les « STANDARD » de programmation.

La plupart des langages utilisés de nos jours dont **Python** sont **multi-paradigmes**.

La **programmation parallèle** est un cas particulier de la programmation concurrente où les opérations s'exécutent toujours au même moment.

Un programme doit être « recetté » ( validé par des tests unitaires et fonctionnels dépendant des spécifications)