

# Synthèse sur les paradigmes

## A. Les différents paradigmes

	Programmation impérative	Programmation fonctionnelle	Programmation orientée objet
<b>Principe général</b>			
<b>Avantage</b>			
<b>Cas d'usage courant</b>			

- Compléter le tableau récapitulatif sur les 3 paradigmes vus dans le cours, en indiquant leur *principe* et leurs *avantages*.
- Associer les différents paradigmes à un cas d'usage/champ d'application parmi la liste ci-dessous :  
*calculs mathématiques - traitement de données - développement de jeu vidéo*

## B. Langages et paradigmes

Les langages informatiques sont souvent pensés pour un paradigme, mais aujourd'hui beaucoup d'entre eux sont multi-paradigmes pour permettre plus de flexibilité et toucher plus de programmeurs. C'est le cas de Python, langage impératif mais pouvant utiliser des caractéristiques fonctionnelles et permettant la POO.

Les langages évoluent avec les pratiques des développeurs. Le succès de la POO a fait qu'elle a été incorporée à des langages conçus à l'origine pour d'autres paradigmes. Par exemple, **C++** est une version "améliorée" du **C** (langage impératif) qui permet d'utiliser la POO. De même, **Cam1** est un langage très fonctionnel, utilisé notamment pour faire des preuves de programmes, qui incorpore la programmation objet dans son évolution **OCam1** ("Objective Cam1").

Plus généralement, il est possible d'intégrer plusieurs langages dans le développement d'un logiciel, pour exploiter le meilleur de chacun.