Calculabilité & Décidabilité

Introduction : Un programme est une donnée

Une donnée est une valeur exploitable dans un programme.
Cette valeur peut être modifiée lors de l'appel de celui-ci.

Cependant, un programme *parent* peut aussi prendre un autre programme comme donnée.
On le nomme programme *enfant*.

```
def cercle(n):
    return 2 * 3.14 * n

def carre(n):
    return 4 * n

def triangle_equilateral(n):
    return 3 * n

def pentagone_regulier(n):
    return 5 * n

def perimetre(forme,n):
    return forme(n)

liste_formes = [cercle,carre,triangle_equilateral,pentagone_regulier]

for forme in liste_formes:
    print(perimetre(forme,5))
```

Utiliser des fonctions comme données peut permettre de modulariser son programme et de l'adapter pour des modifications futures (ajout de fonctions, modification de fonctions déjà existantes...).

Définitions

Un algorithme est une suite **finie** d'instructions qui résout un problème.

Un problème en informatique est une traduction mathématique d'une question que l'on peut se poser.

On dit qu'un problème est calculable si ce problème peut être modélisé par un algorithme.
On appelle **théorie de la calculabilité** l'étude de la calculabilité d'un problème, c'est-à-dire, si un problème est calculable ou non.

On dit qu'un problème est décidable si ce problème peut être répondu par oui ou non, True ou False.
On appelle **théorie de la décidabilité** l'étude de la décidabilité d'un problème, c'est-à-dire, si un problème est décidable ou non.

La calculabilité et la décidabilité sont des fondements de l'informatique théorique.

L'origine de l'interêt de ces théories remonte au XVIIème siècle quand Gottfried Wilhelm Leibniz, un érudit, se demandait s'il était possible qu'une machine puisse résoudre tous les problèmes (mathématiques, de décision, de calcul...).

Cette reflexion a été remise au premier plan en 1928 par les mathématiciens allemands David Hilbert et Wilhelm Ackermann en posant le *Entscheidungproblem* ou *problème de décision* : "Peut on toujours décider de façon mécanique si un énoncé est vrai?"

Est-ce que tout est calculable ?

Machine de Turing

Lambda Calcul

```
pentagone_regulier = lambda x : 5*x
```

Une manière pour démontrer : Problème de l'arret