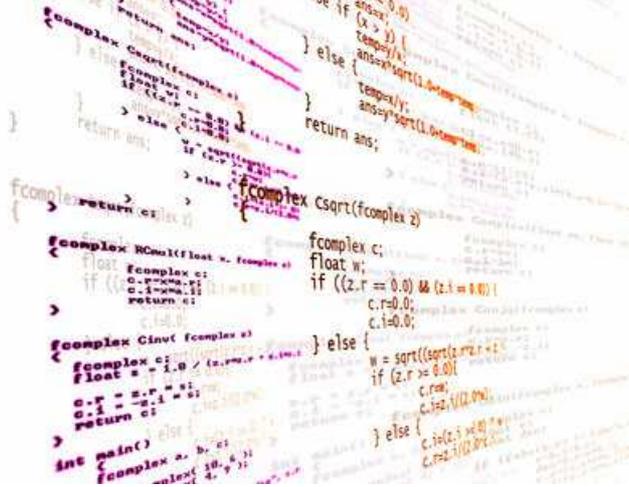


# Introduction à la programmation



## Baccalauréat S - Spécialité ISN

- 4.3 : Langues et programmation

### [Référentiel de formation](#)

Créé avec HelpNDoc Personal Edition: [Écrire des livres électronique Kindle](#)

## Objectifs

A la fin de cette séquence l'élève est capable :

- d'expliquer la différence entre les types de langages,
- de classer les niveaux de langages
- d'utiliser le vocabulaire de base utilisé en programmation

## Prérequis

[Codage de l'information](#)

## Programmation

### Introduction

La programmation est la science qui permet d'élaborer un programme. Dans le sujet qui nous intéresse, il s'agit ici de **Programme Informatique** c'est à dire de programmes destinés à être utilisés avec des outils numériques modernes.

L'élaboration d'un programme informatique nécessite toujours au préalable une réflexion sur la compréhension du problème. Cette réflexion donnera lieu à des choix de technologie et de codage. Cette réflexion conditionnera l'efficacité du programme.

**Codage** : C'est l'action qui consiste à écrire le programme.

Un programme informatique est développé de manière très différente selon la technologie employée mais il est toujours composé de constantes, de variables et d'une suite d'**instructions** destinées à être exécutées de

manière automatique par un appareil capable de gérer de l'information. Un **logiciel** est un ensemble composé de plusieurs programmes et des fichiers nécessaires à son fonctionnement.

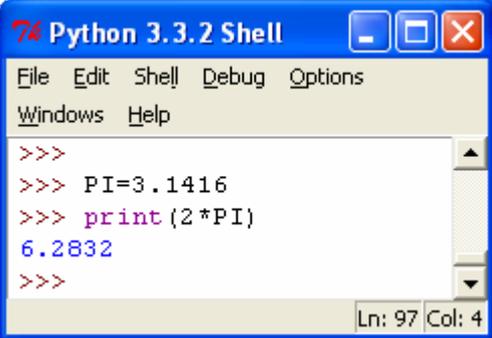
Pour structurer la programmation et gérer des projets complexes, des outils tels que les algorithmes sont utilisés.

## Programmation : Définitions

### Constantes informatiques

Une constante est une valeur qui ne change pas dans le programme ou une partie du programme.

**Par exemple** : le nombre PI, s'il est utilisé dans un programme de calcul de surface ne changera pas. Plutôt que de réécrire plusieurs fois dans un programme la valeur de PI (3,1417...), on peut initialiser une constante appelée PI et utiliser par la suite cette constante.

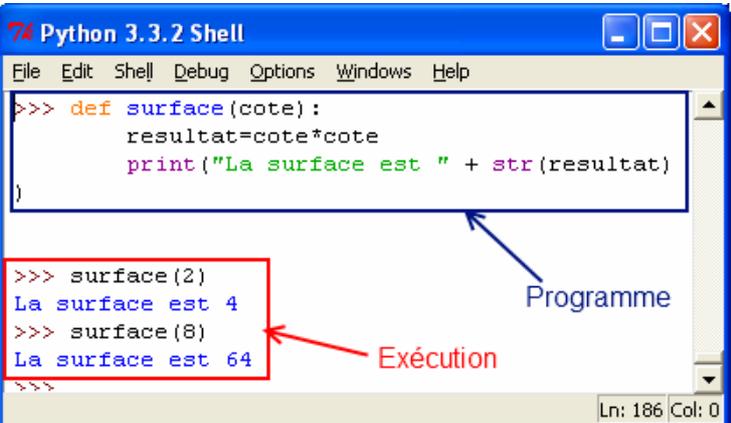


```
Python 3.3.2 Shell
File Edit Shell Debug Options
Windows Help
>>>
>>> PI=3.1416
>>> print(2*PI)
6.2832
>>>
```

### Variable informatique

Une variable est une valeur amenée à changer lors de l'exécution du programme.

**Par exemple** : Dans un programme destiné à calculer une surface d'un carré, la valeur du côté changera à chaque nouveau calcul. Cote sera donc une variable.



```
Python 3.3.2 Shell
File Edit Shell Debug Options Windows Help
>>> def surface(cote):
>>>     resultat=cote*cote
>>>     print("La surface est " + str(resultat))
>>>
>>> surface(2)
La surface est 4
>>> surface(8)
La surface est 64
>>>
```

Annotations: "Programme" points to the function definition, "Exécution" points to the function calls.

### Type

Une constante ou une variable peuvent être un caractère, une chaîne de caractère, un entier, un nombre décimal (flottant) etc... On parle du type de la variable.

La plupart des langages informatiques imposent de définir le type des variables utilisées.

#### Exemple de types en C++ :

- int i, j, k;
- char c, ch;
- float f, salary;
- double d;

### Fonction

Pour pouvoir gérer la complexité d'un problème, il est nécessaire de simplifier celui-ci en le décomposant en un ensemble de petits problèmes.

La fonction permet de faire cela. Une fonction prend généralement en compte des paramètres et restitue un résultat.

**Par exemple :** la fonction qui restitue le carré d'une variable passée en paramètre est réutilisée ailleurs dans le programme

Dans la plupart des langages informatiques, le mot clé pour définir une fonction est : **FUNCTION**

## Procédure

L'objectif de la procédure est identique à celui de la fonction mais une procédure ne rend pas de résultat à proprement parlé. Elle réalise une action : affichage à l'écran, action sur un périphérique, initialisation de variables internes au programme, etc....

**Dans l'exemple** ci-contre, la procédure sauvegarde la configuration du programme.

```

Python 3.3.2 Shell
File Edit Shell Debug Options Windows Help
>>> def carre(x):
>>>     return x*x
>>> carre(2)
4
>>> def equation(x):
>>>     return carre(x)+x+1
>>> equation(2)
7
>>> |

Delphi 7 - MOTO6811
Fichier Edition Chercher Voir Projet Exécuter Composant Base de données Outils Fenêtre Aide | dactnet
Standard | Supplément | Win32 | Système | AccèsBD | ContrôleBD | dB
MOTO6811 M6811
procedure TWinMain1.Sauver1Click(Sender: TObject);
(sauver les configurations)
var
  ini:TRegIniFile;
begin
  ini:=TRegIniFile.create(IniFile);
  try
    (on sauve la police de caractères)
    Ini.WriteString('POLICE','Font',uPWin.Font.Name);
    Ini.WriteInteger('POLICE','Taille',uPWin.Font.Size);
    Ini.WriteString('POLICE','Couleur',INTTOSTR(uPWin.Font.Color));
    //On sauve les fenêtres
    Ini.WriteBool('Bureau','Programme',Chk_Win_Prg);
    Ini.WriteBool('Bureau','ROM',Chk_Win_ROM);
    Ini.WriteBool('Bureau','RAM',Chk_Win_RAM);
    Ini.WriteBool('Bureau','INTF',Chk_Win_Intf);
    If Chk_Win_Prg then
      begin
        Ini.WriteInteger('Bureau','Prg_X',WinPrg.Left);
        Ini.WriteInteger('Bureau','Prg_Y',WinPrg.Top);
      end;
  end;

```

## Variables et constantes locales

Le plus souvent il est possible de définir des variables ou des constantes dans les fonctions ou les procédures. En dehors de celles-ci, les variables ou constantes n'ont pas d'existence dans le programme. On parle de **variable ou constantes locales**.

## Variables et constantes globales

Lorsque les variables ou constantes sont définies en dehors des fonctions ou procédures elles sont accessibles partout dans le programme.

Lorsqu'une variable locale porte le même nom qu'une variable globale, la variable globale est ignorée dans la procédure.

**Dans l'exemple** ci-contre, \$a et \$b sont des constantes globales mais des variables locales du même nom existent dans la fonction somme. Lors de l'exécution du programme les constantes globales sont ignorées.

```

test.php
1 <?php
2 $a=10; //variable globale
3 $b=2; //variable globale
4 function somme ($v1,$v2)
5 { $a=$v1; //variable locale
6   $b=$v2; //variable locale
7   return $v1+$v2;}
8 echo "la somme de a+b est égale à ".somme(4,2);
9 ?>

Résultat
lyc-deck-gl x SILOGED x Boite
127.0.0.1/test.php
la somme de a+b est égale à 6

```

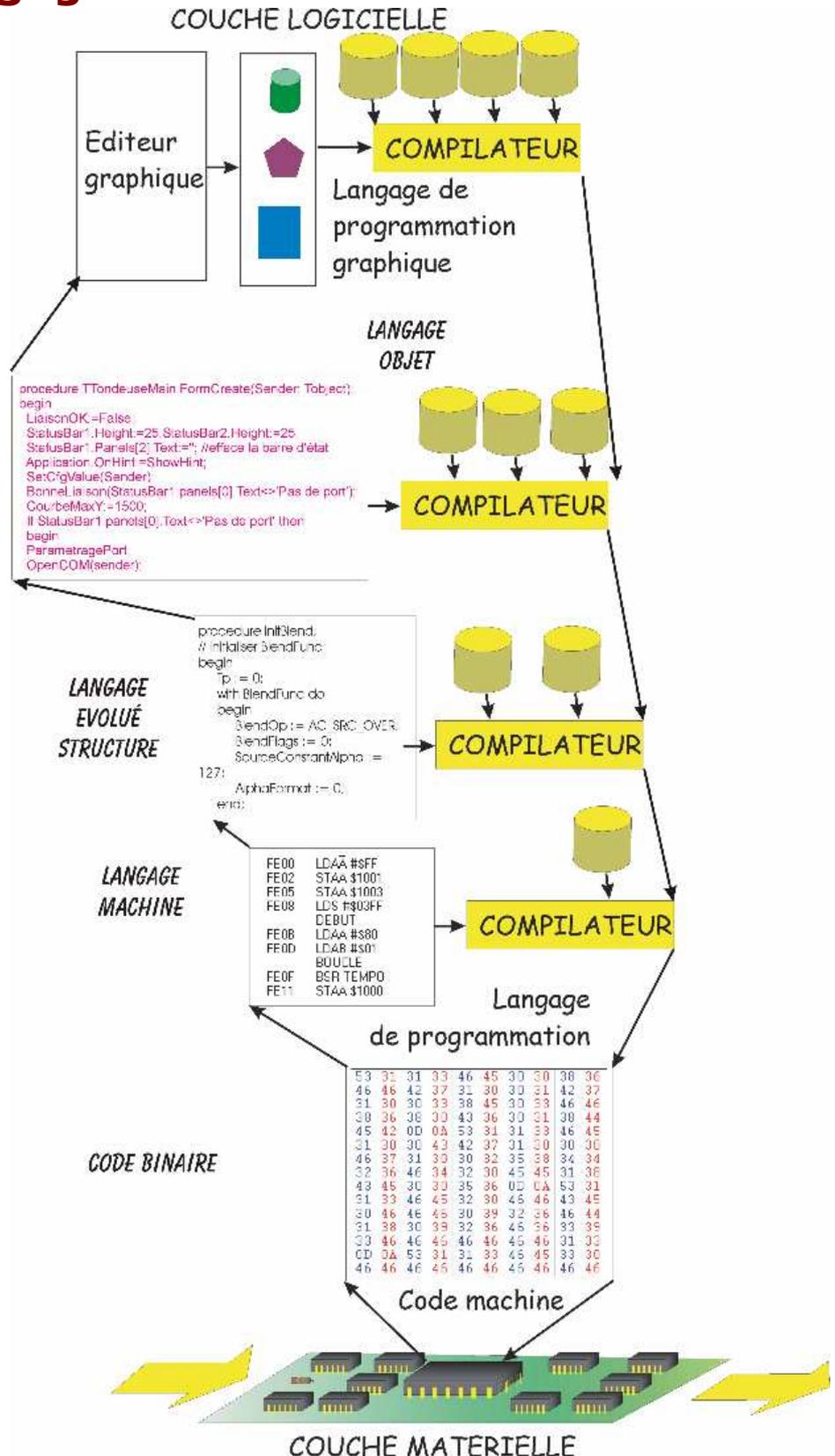
# Langages de programmation

Tout programme fini toujours par être implanté dans un composant électronique. Et doit donc toujours finir par un ensemble d'informations binaires.

## Niveaux de langage

Les instructions comprises par le microprocesseur sont codées sous forme de mots binaires. Le programme composé de ces mots est appelé **CODE MACHINE**. Il est implanté dans les mémoires de l'ordinateur.

Pour rendre la programmation plus facile et plus rapide - pour éviter la programmation par code machine- de nombreux langages informatiques sont nés. Le **langage informatique** est un langage formel permettant de décrire un problème lié au traitement de l'information. Plus le langage est proche de la structure matérielle et plus le programme est compact. Plus on s'éloigne de la couche matérielle et plus on se dédouane des contraintes matérielles, la programmation est plus rapide, mais les programmes deviennent plus gros (en nombre d'octets).



# Langages : Définitions

## Compilateur

La suite d'instructions dans un langage programme est appelé **CODE SOURCE**.

L'outil logiciel permettant de transformer le code source en code binaire est appelé **COMPILATEUR**.

## Langage compilé

On parle de langage compilé lorsque le code source passe par un compilateur et que le programme peut-être exécuté sans logiciel supplémentaire (si ce n'est le système d'exploitation de l'ordinateur).

## Langage interprété

A l'inverse, un langage interprété nécessite un logiciel supplémentaire pour que le programme s'exécute. Les pages WEB que l'on consulte sur Internet sont interprétées par le navigateur WEB.

## Assembleur

Le terme assembleur a deux significations :

- C'est le langage très proche du microprocesseur composé d'instructions élémentaires codés en mnémoniques.
- c'est l'outil logiciel qui transforme le code assembleur en code machine

## Langages objet

Ce sont des langages informatiques de haut niveau pour lesquels des objets visuels ou comportementaux sont associés à des méthodes et à des propriétés. Ces objets héritent des caractéristiques de leurs parents et ils peuvent transmettent leurs caractéristiques à leurs enfants. Un leçon spécifique de la formation ISN traitera de la programmation objet.

# Quelques langages informatiques

### Basic

Initialement le basic était un langage interprété livré avec son interpréteur GWBasic dans les premiers IBM-PC. Il se voulait simple et facile à utiliser. GWBasic n'était pas un langage structuré. Son successeur Visual Basic l'est devenu.

Microsoft, a développé une version compilé de Visual Basic (VB) mais ce langage existe encore en version interprété.

### Cobol (COmon Business Oriented Language)

Langage de programmation très ancien (1959) spécialisé dans la programmation de gestion. Il est encore utilisé car il est difficile de remplacer des codes écrits pour toutes les banques du monde.

### Fortran (FORmula TRANslator)

Langage ancien également utilisé principalement dans les calculs

scientifiques. Il est maintenant à l'abandon et remplacé par le C++

### **Pascal**

Pascal a été développé au départ pour l'enseignement de l'Informatique. C'est un langage structuré d'une grande rigueur. Il est peu utilisé maintenant. Le langage VHDL a repris partiellement les structures du Pascal.

### **Delphi**

Delphi est une marque déposée de Borland. Il s'agit de la version Objet de Turbo Pascal.

### **C**

C est un langage structuré très utilisé en programmation. Le PHP et Java ont repris les structures du C.

### **C++**

C'est la version objet du C.

### **HTML (HyperText Markup Language)**

Langage interprété utilisé pour la programmation des pages WEB

### **PHP (Hypertext PreProcessor)**

Langage utilisé pour la programmation WEB. Les structures viennent du C.

### **ASP (Active Server Page)**

Développé par Microsoft pour concurrencer PHP

### **Java**

Langage orienté fortement objet reprenant la syntaxe de C++ mais duquel les pointeurs ont été retirés. Les applications Java sont surtout utilisées dans des usages client-serveur donc en réseau.

### **Python**

C'est un langage objet de licence libre et multi plateformes.

## **Notions de droits**

Quand on écrit un programme qui est utilisé par d'autres personnes que soi, on doit préciser ses conditions d'utilisation par un contrat qui s'appelle une licence logicielle.

Une installation et/ou utilisation de logiciel ne peut se faire qu'après acceptation et respect des conditions générales d'utilisation du logiciel (CGU).

Les licences d'utilisation sont multiples et très variées :

### **Licence commerciale**

Ce sont typiquement les logiciels que vous achetez. Parfois ces logiciels ne sont que loués (antivirus par exemple) car leur utilisation n'a aucun sens sans mise à jour et c'est ma mise à jour qui est payante. L'acceptation de la licence d'utilisation sous entend généralement l'installation et l'utilisation sur un seul poste informatique. L'utilisateur est autorisé à faire une copie du support d'installation.

### **Domaine public**

Un logiciel sous licence de type « domaine public » est un logiciel pour lequel l'utilisateur, outre la possibilité de redistribuer et de modifier le logiciel, est autorisé à ajouter des restrictions à la licence sous laquelle il redistribue le logiciel (certaines copies, ou certaines versions modifiées, pouvant de facto ne pas être libres du tout et devenir alors propriétaire).

### **Shareware (Partagiciels)**

Attribue un droit temporaire et/ou des fonctions limitées du logiciel. Après une période d'essai, l'utilisateur doit rétribuer l'auteur ou arrêter d'utiliser le logiciel.

### **Freeware**

C'est un logiciel que l'auteur distribue gratuitement mais généralement il ne livre pas les sources et l'utilisateur est tenu malgré tout de respecter les CGU.

### **GPL(GNU)**

GNU est un projet collectif visant à contourner l'usage commercial d'Unix en développant un UNIX gratuit. GNU Vient de Gnu is Not Unix.

La Licence Générale Publique de GNU, ou General Public License, est la licence principalement utilisée pour le développement de logiciels libres. Les utilisateurs de code GPL sont libres d'exécuter le code, de le modifier et de distribuer ces modifications, à condition que ces modifications soient aussi sous licence GPL.

### **Open Source**

Quand un logiciel est opensource, on peut accéder au code source du programme et voir son fonctionnement interne. Généralement les licences GNU sont OpenSource.