



# ***Systemes et Technologies d'Information Initiation***

***Mathématiques 3ème  
Module No 22***

## ***Concepts de programmation***



# Objectifs généraux



Qu'est-ce qu'une  
machine  
programmable ?

Qu'est-ce qu'un  
langage ?



# Le réseau informatique, outil idéal du S.I.

- En plus de ses possibilités de Collecte, de Stockage, de Traitement et de Communication, l'ordinateur est une **machine programmable**.
- Ses possibilités de **Programmation**, c'est à dire d'adaptation à un problème nouveau, permettent de développer des **logiciels** spécifiques capables de satisfaire aux besoins exprimés par les utilisateurs.
- Ils permettent aussi, lorsque le marché est suffisamment vaste pour justifier un tel investissement, à des sociétés spécialisées de développer des solutions standards : les **progiciels**.



# Programmation et programmation

## → **Programmation**

- Traduction d'un problème dans un langage compréhensible par l'ordinateur.

## → **Programmation**

- Méthode pour concevoir un bon programme

## → La plupart des langages informatiques sont **procéduraux**. Ils imposent de décomposer la solution au problème posé sous forme d'un **algorithme** (Voir leçon 1)

## → Seuls quelques langages de 4ème génération, intégrant les technologies de l'intelligence artificielle, sont **assertionnels**.

## → SQL, langage d'interrogation des bases de données est assertionnel.



# Génération de langage

## PREMIERE GENERATION

→ **Langage de Base** : Le langage binaire que les circuits de l'ordinateur peut décoder

## SECONDE GENERATION

→ **Langage Symbolique** : S'affranchir des contraintes du codage en binaire

## TROISIEME GENERATION

→ **Langage Evolué** : S'affranchir des contraintes de la spécificité du répertoire d'instructions

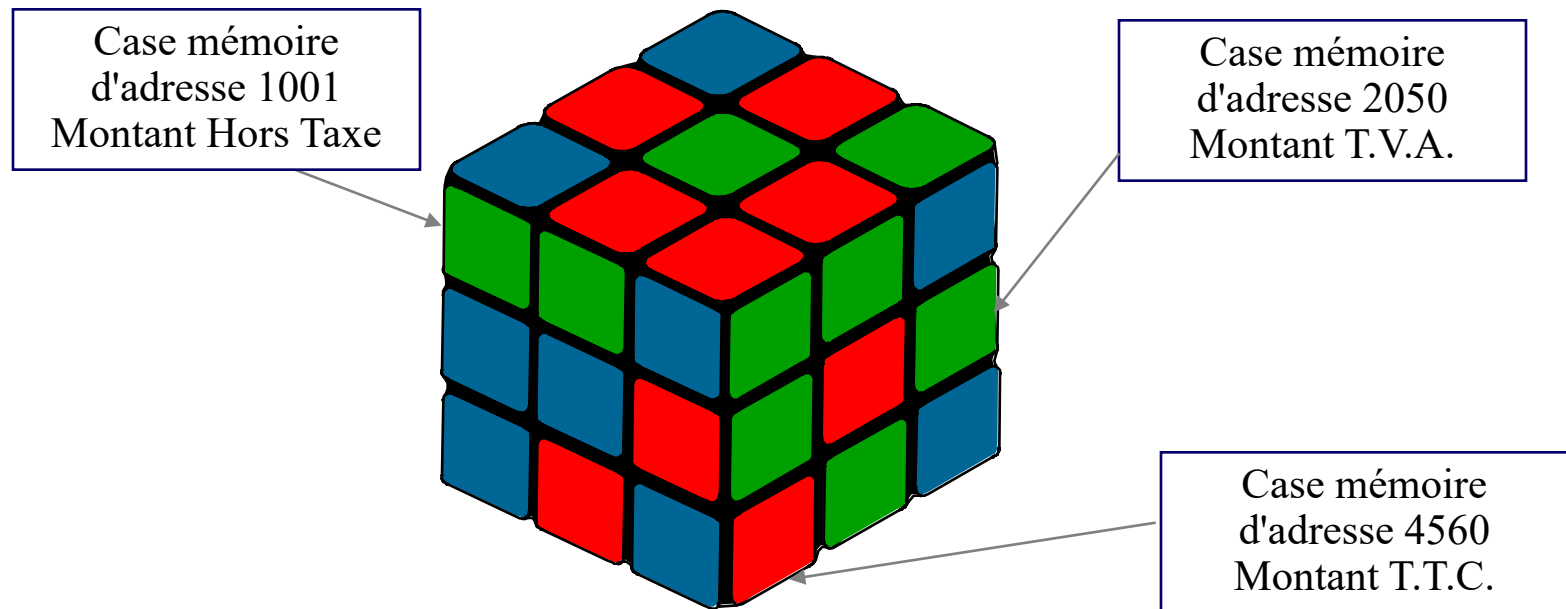
## QUATRIEME GENERATION

→ **Langage "de quatrième génération"** : Se rapprocher du langage naturel

Une typologie commode, mais qui ne signifie pas que seuls les langages de 4 ème génération sont utilisés aujourd'hui !

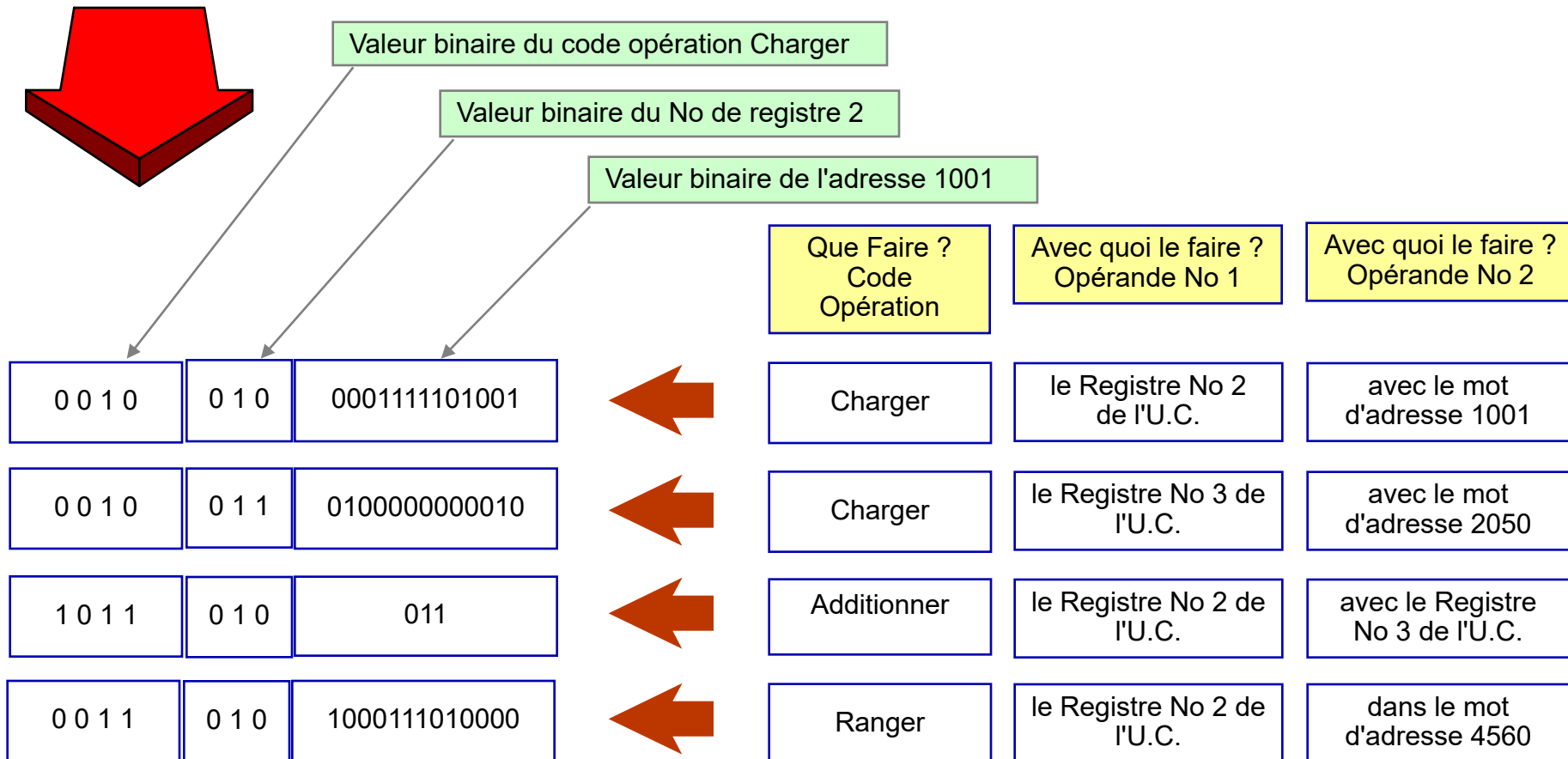
# → Principes de la programmation

- Nous nous proposons de rappeler les principes des langages de programmation en tentant de résoudre un problème très simple
- Effectuons la somme d'un opérande stocké à l'adresse 1001 (le montant Hors Taxe) avec un opérande stocké à l'adresse 2050 (le montant de la T.V.A.).
- Le résultat calculé (le montant T.T.C.) doit être stocké à l'adresse 4560.



# → Ce que comprend l'ordinateur

- Chaque ligne du programme correspond à une instruction du **répertoire des instructions de la machine** et est codée directement dans le langage binaire interprétable par l'ordinateur.



# → Ce que comprend l'ordinateur

## → Le programme 1G

- 00100100001111101001
- 00100110100000000010
- 10110100000000000011
- 00110101000111010000

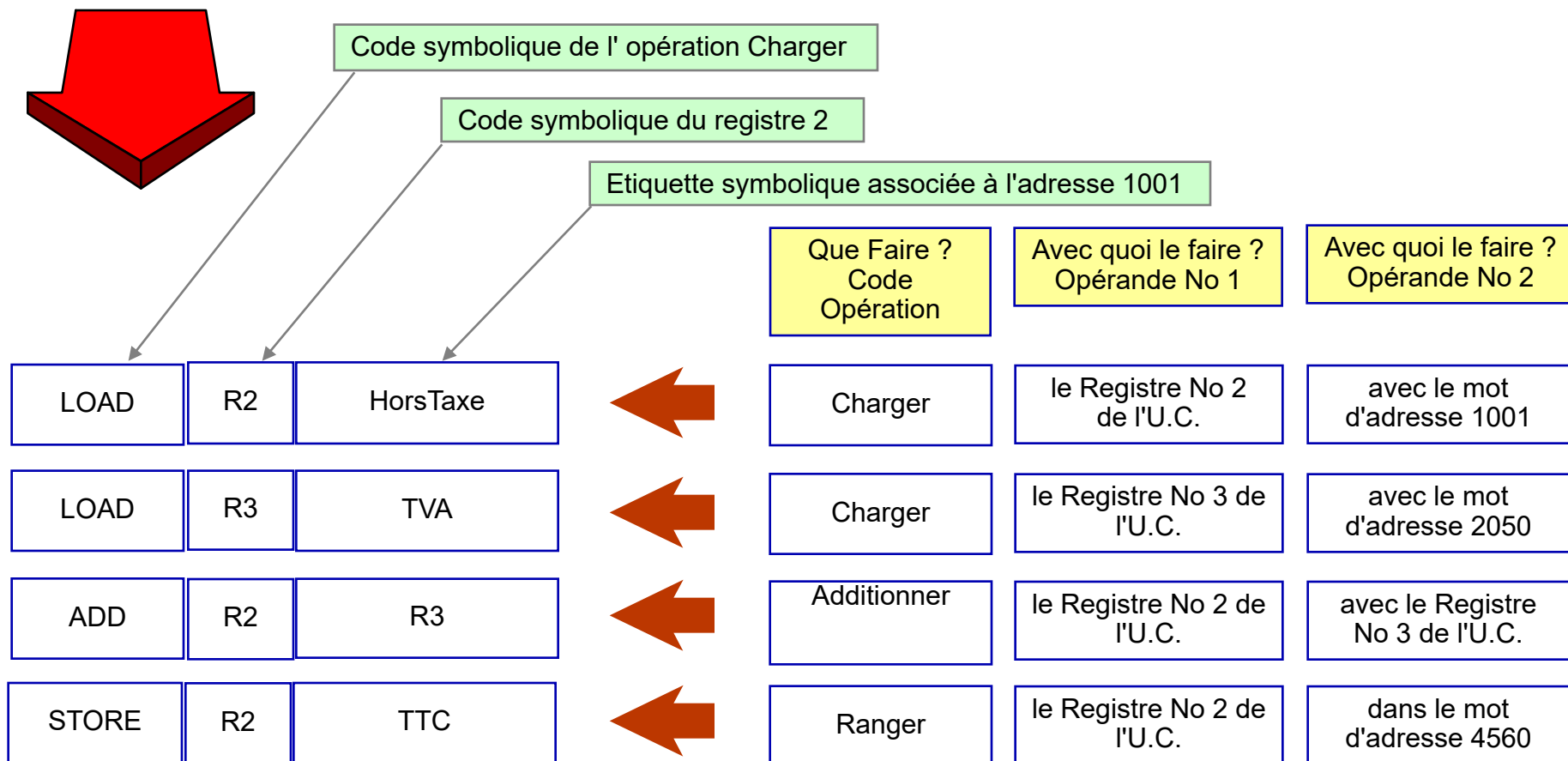


- Seul l'ordinateur le comprend
- Il peut ne comprendre que ce langage
- Il se programmait avec les clefs du panneau de commande
- Il se lisait avec les voyants du panneau de commande



# → Deuxième génération

- A chaque élément de l'instruction est associée un équivalent symbolique, ce qui en facilite l'écriture, mais à chaque ligne du programme correspond toujours une et une seule instruction du répertoire.





# Deuxième génération

- Le programme 2G
  - LOAD R2, HorsTaxe
  - LOAD R3, TVA
  - ADD R2, R3
  - STORE R2, TTC

```

Apple IIe Emulator
82
      STA  _MODE_HGR
      RTS

;=====
; Enclenche mode graphic
;=====

_SET_GRAPHIC JSR  CLR_LINE
             NOP
             NOP
             LDA  _MODE_HGR
             BPL  HBF3E+1 ; BUG!!!! must be $BF3D

             STA  TXTCLR ; graphic
             STA  HIRES  ; mode HGR
             LDA  _PAGE_HGR
             BPL  HBF3E

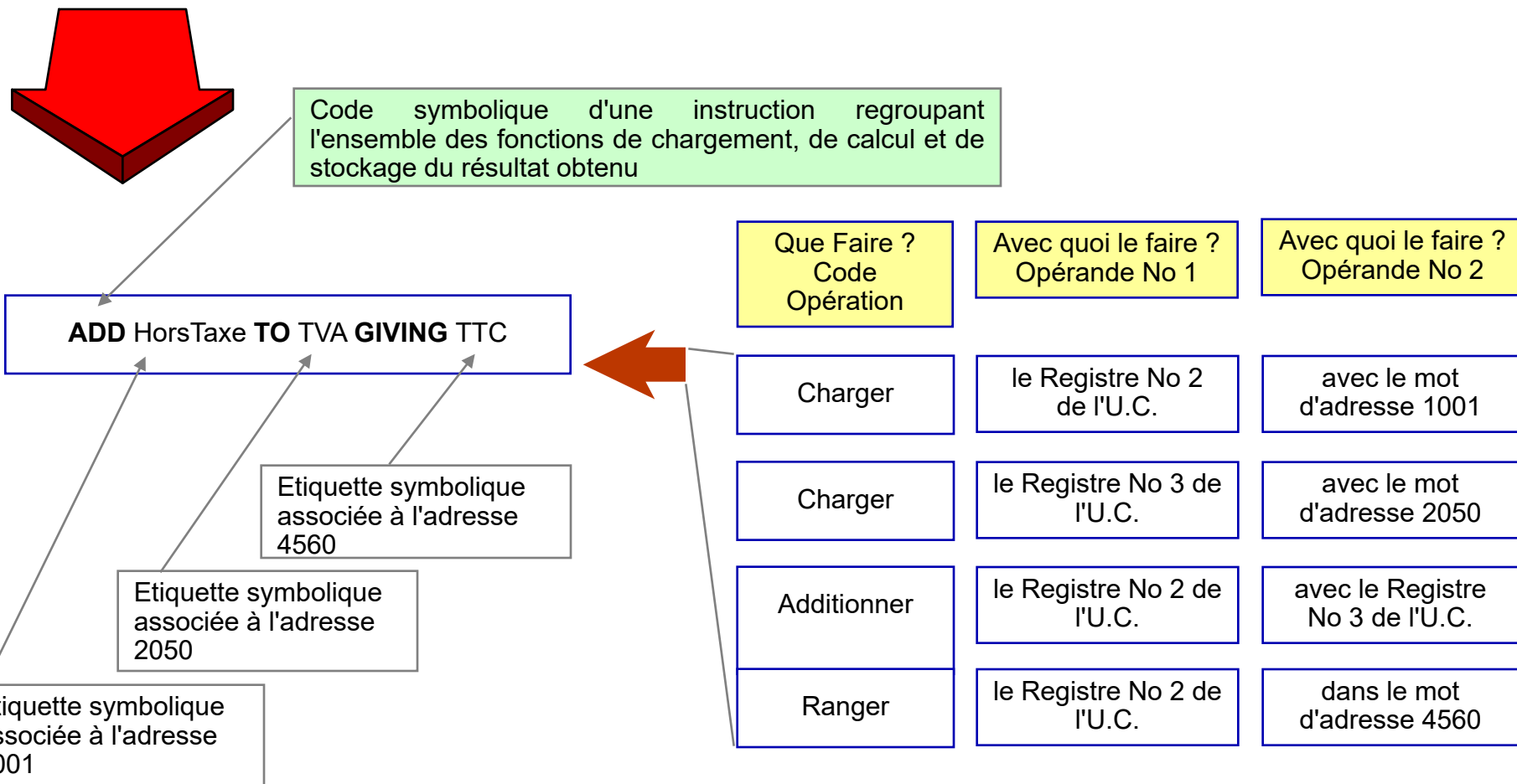
HBF3D     STA  HISCY ; HGR2
             RTS

HBF3E     STA  LOWSCR ; HGR1
             RTS
  
```

- Comme l'ordinateur ne comprend que le langage 1G, ce programme doit être traduit.
- Ce traducteur est un programme **Assembleur**
- Un langage 2G Assembleur est spécifique d'une machine donnée.

# → Troisième génération

- Le principe des étiquettes symboliques est conservé, et chaque ligne du programme engendre plusieurs instructions du répertoire.





# Troisième génération

- Le programme 3G
  - ADD HorsTaxe TO TVA  
GIVING TTC

```
SCONSTO.M214.LIBRARY(MSGTSTCO) - 01.07          Columns 00001
> Scroll ==>
do-the-work.
  move in-part-number      to reprt-part-number
  move in-description      to reprt-description
  move in-quantity-on-hand to reprt-quantity-on-hand
  move in-quantity-on-ord  to reprt-quantity-on-ord
  move in-unit-price       to reprt-unit-price
  move in-reorder-level    to reprt-reorder-level
  write reprt-rec from reprt-record
  perform read-a-record.

print-table.
  move parts-no(part-index) to reprt-part-number
  move parts-desc(part-index) to reprt-description
  move parts-on-hand(part-index) to reprt-quantity-on-hand
  move parts-on-ord(part-index) to reprt-quantity-on-ord
```

- Comme l'ordinateur ne comprend que le langage 1G, ce programme doit être traduit.
- Ce traducteur est un programme **Compilateur**
- Un langage 3G est indépendant de la machine mais un compilateur est dépendant de la machine.



# Quatrième génération

- Les Langages dits "**de Quatrième Génération**" (L4G) ont pour vocation de s'affranchir du caractère hermétique des langages informatiques classiques.
- Cependant, le développement d'applications s'est avéré un domaine trop complexe pour autoriser l'utilisation d'un langage très proche du langage naturel, par nature trop imprécis et trop ambigu.
- Certes, les L4G de ce domaine ont apporté des gains de lisibilité et de productivité importants, mais ne sont que de super langages évolués.
- Par contre, les langages spécialisés dans le domaine de l'interrogation des Bases de Données, domaine plus simple parce que bien délimité, ont tenté avec plus de succès de se rapprocher du langage naturel.



# Quatrième génération

## → Le programme 4G (SQL)

SELECT

Date

FROM

Voitures, Etudiants, Posseder

WHERE

Prenom="Larfi"

AND Marque="VW Golf"

AND Etudiants.Numero = Posseder.Numero

AND Voitures.NumeroV = Posseder.NumeroV;



# Passer du langage symbolique au langage binaire

- Nécessité d'un **traducteur** pour passer d'un Langage Symbolique ou Evolué au Langage de Base, seul compréhensible par les circuits de l'ordinateur.
- Notion de **Programme Source** (avant traduction) et de **Programme Objet** (après traduction).
- Pour passer du Langage Evolué au Langage de Base en créant et en conservant un programme objet : le **Compilateur**.
- Pour passer du Langage Evolué au Langage de Base pour une exécution immédiate, sans conserver la programme objet : **l'Interpréteur**.
- Pour passer du Langage Symbolique au Langage de Base : le programme **Assembleur**.



# Exemples



- Des milliers de langages informatiques
- Les plus connus :
  - Fortran
  - Cobol
  - Basic
  - C
  - Java
  - Python





# Autres langages

- A côté des **langages de programmation** dont nous venons de parler et qui implantent physiquement les modèles de données et de traitements traduits en algorithmes, il y a d'autres langages :
- des **langages de définition de données** qui rendent concrets les modèles conceptuels de données ;
- des **langages de requête**, comme SQL, qui permettent de réaliser des traitements (création, interrogation, mise à jour, suppression) sur les données définies;
- des **langages de description de pages**, propres au monde du web, comme HTML ou XML, qui utilisent des balises insérées dans le texte pour le structurer et lui donner une forme particulière.



# Cas d'une application web

- Une application web peut combiner plusieurs de ces langages :
  - **HTML** (description) pour définir le contenu et la forme des pages avec diverses balises;
  - **Javascript** (programmation) pour animer la page en exécutant au niveau du poste de l'utilisateur (client) du code inséré dans la page entre des balises `<SCRIPT>` et `</SCRIPT>`;
  - **PHP** (programmation) pour animer la page en exécutant au niveau du serveur du code inséré dans la page entre des balises `<?php` et `?>`;
  - **SQL** (requête) qui sera appelé par le code PHP pour interroger la base de données.