

Analyse et programmation 1

Introduction du langage C

Introduction au langage C

Objectif de ce cours

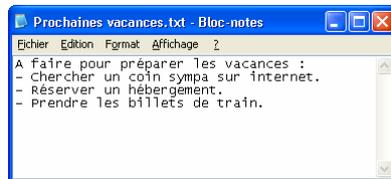
- Découvrir le langage C
 - Son histoire.
 - Ses forces.
 - Ses faiblesses.
- Montrer comment créer un programme.
 - Comment écrire un programme.
 - Comment le faire fonctionner.
- Analyser l'intérêt de cet apprentissage.

heig-vd

Rappel

Qu'est ce qu'un programme

- Pour l'utilisateur
 - Un programme doit rendre des services.
 - Il doit répondre aux commandes de l'utilisateur.
 - Il permet de créer, transformer et enregistrer de l'information.
 - Exemple : le programme Bloc-notes (notepad.exe)
 - Taper un texte dans un éditeur : création.
 - Impression du texte : transformation.
 - Sauvegarde dans un fichier : enregistrement.



heig-vd

Rappel

Qu'est ce qu'un programme

- Pour la CPU
 - Un programme est une suite de nombres, représentés en binaire, et circulant sous forme électrique.
 - Certains nombres sont des numéros d'instruction.
 - D'autres nombres sont des opérandes.
- Exemple de programme
 - Le bloc notes (notepad.exe)

L'outil DOS « debug » procède à la lecture du fichier. Ensuite, il affiche le code en LANGAGE MACHINE et sa traduction symbolique (assembleur), répartie suivant les Colonnes

ADRESSE	CONTENU	INSTR.	OPERANDES
---------	---------	--------	-----------

```
C:\>debug C:\Windows\notepad.exe
-u
15B0:0000 0E          PUSH   CS
15B0:0001 1F          POP    DS
15B0:0002 BA0E00     MOV    DX,000E
15B0:0005 B409     MOV    AH,09
15B0:0007 CD21     INT    21
15B0:0009 B8014C     MOV    AX,4C01
15B0:000C CD21     INT    21
15B0:000E 54          PUSH   SP
15B0:000F 68          DB     68
15B0:0010 69          DB     69
15B0:0011 7320     JNB    0033
15B0:0013 7072     JO     0087
15B0:0015 6F          DB     6F
15B0:0016 67          DB     67
15B0:0017 7261     JB     007A
15B0:0019 6D          DB     6D
15B0:001A 206361   AND    [BP+DI+61],AH
```

heig-vd

Rappel

Mission du programmeur : créer les programmes

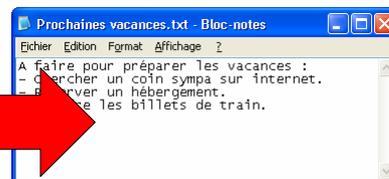
- Le programmeur doit créer la suite de nombres
 - Qui représente la séquence des opérations à effectuer.
 - Cette séquence doit réaliser les fonctions souhaitées par l'utilisateur.
- Problèmes
 - Les instructions de la CPU sont très élémentaires.
 - La programmation d'un calcul mathématique tout simple est déjà très compliquée.
 - La création d'applications réelles selon cette approche semble donc très difficile.
 - Absence de portabilité
 - Les numéros d'instruction sont spécifiques à chaque microprocesseur.
 - Un programme écrit pour un microprocesseur devrait être complètement réécrit pour être réutilisé avec un autre.

heig-vd

Rappel

Mission du programmeur : créer les programmes

```
C:\>debug C:\Windows\notepad.exe
-u
15B0:0000 0E          PUSH  CS
15B0:0001 1F          POP   DS
15B0:0002 BA0E00      MOV   DX,000E
15B0:0005 B409      MOV   AH,09
15B0:0007 CD21      INT   21
15B0:0009 B8014C      MOV   AX,4C01
15B0:000C CD21      INT   21
15B0:000E 54          PUSH  SP
15B0:000F 68          DB    68
15B0:0010 69          DB    69
15B0:0011 7320      JNB   0033
15B0:0013 7072      JO    0087
15B0:0015 6F          DB    6F
15B0:0016 67          DB    67
15B0:0017 7261      JB    007A
15B0:0019 6D          DB    6D
15B0:001A 206361    AND   [BP+DI+61],AH
```



heig-vd

Pourquoi le langage C ?

Motivations de départ

- Le langage C a été inventé :
 - Par les concepteurs du système d'exploitation Unix en 1972.
 - Pour faciliter leur travail de programmation.
- Principes du langage C
 - Un programme C est écrit sous une forme symbolique.
 - On utilise des expressions symboliques comme « $x = y + 5$; »
 - Il est constitué d'un texte humainement intelligible.
 - Les mots utilisés sont issus de la langue anglaise.
 - Il n'est pas écrit pour un microprocesseur particulier.
 - Contrairement au langage machine.

heig-vd

Pourquoi le langage C ?

Avantages

- Meilleure facilité de programmation
 - Ce langage permet de créer des applications de complexité réelle.
 - Permet de programmer sans connaître le langage machine de la CPU.
 - Cependant, ce langage reste assez proche du matériel
 - Utilise la notion d'adresse.
 - `&x`, `*px`
 - Possède des opérations permettant de manipuler le binaire
 - champs de bits, `<<`, `>>`, `&`, `|`, `^`
 - Permet l'écriture hexadécimale et octale quand c'est nécessaire.
 - `0xB0B0`, `'\034'`
 - Très efficace à l'exécution.
- Portabilité
 - Un programme en langage C peut être exploité sur tout type de CPU
 - Moyennant cependant quelques précautions !

heig-vd

Un mini-exemple commenté

Le programme Hello World !

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Hello World !\n");
    system("PAUSE");
    return 0;
}
```

inclusion des fichiers «stdio.h» et «stdlib.h» qui définissent «printf» et «system»

début du programme. « main » veut dire principal en anglais.

L'accolade ouvrante { marque le début d'un groupe d'instructions.

printf affiche un texte ou une valeur à l'écran. « \n » écrit un saut de ligne.

Cette instruction attend qu'une touche soit pressée

« return » termine l'exécution de main. Elle renvoie souvent une valeur, ici 0.

L'accolade fermante } marque la fin d'un groupe d'instructions.

heig-vd

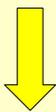
Un mini-exemple commenté

Analyse

- Un programme en langage C
 - Se déroule ou s'exécute séquentiellement
 - n'est pas directement exécutable par la CPU.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Hello World !\n");
    system("PAUSE");
    return 0;
}
```



heig-vd

Comment rendre exécutable un programme C ?

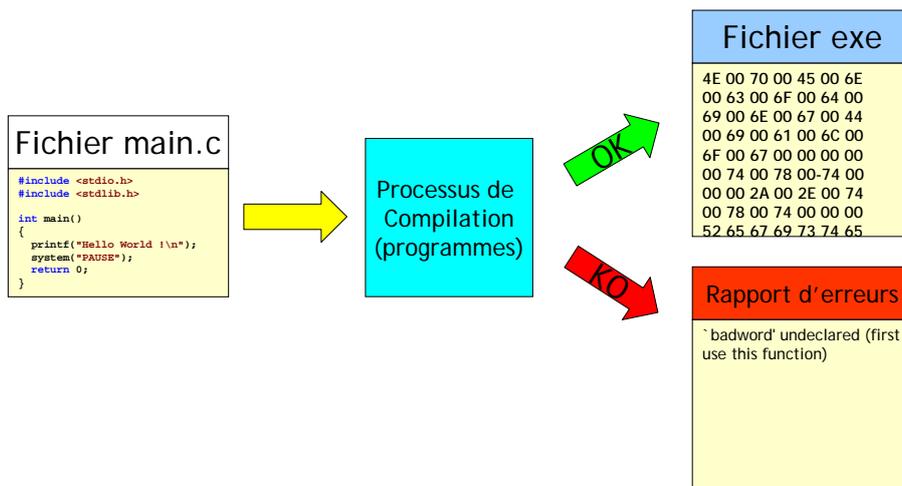
Une étape de traduction

- Pour devenir exécutable, un programme en langage C
 - Doit être traduit en langage machine.
 - Cette opération est fastidieuse est très répétitive.
 - Heureusement, elle est faite automatiquement par un programme.
- La traduction en langage machine
 - Cette opération s'appelle la « compilation ».
 - Elle est faite par un programme appelé « Compilateur »
 - C'est une opération assez complexe en plusieurs étapes.
 - Au cours de la compilation
 - Le code source en langage C est vérifié.
 - Les erreurs grossières sont signalées au programmeur.
 - Si tout a bien marché
 - le résultat est un nouveau fichier.
 - Ce fichier est un programme en langage machine, exécutable.

heig-vd

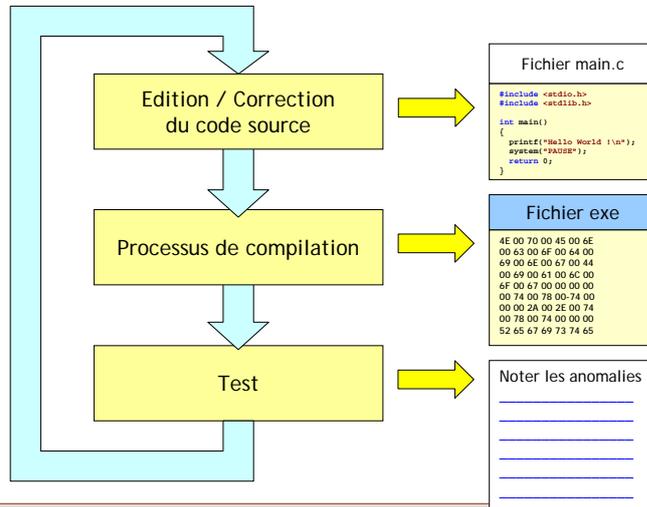
Comment rendre exécutable un programme C ?

La compilation - le principe



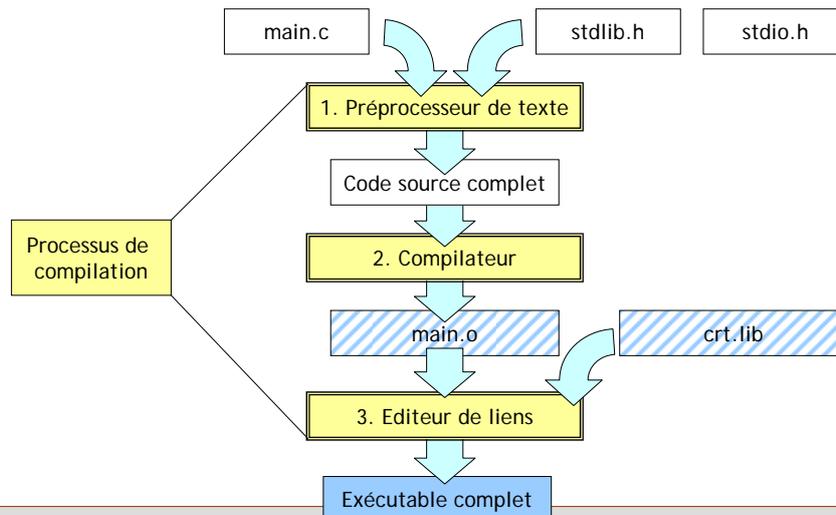
Comment rendre exécutable un programme C ?

Modèle du cycle de développement



La compilation, un processus complexe

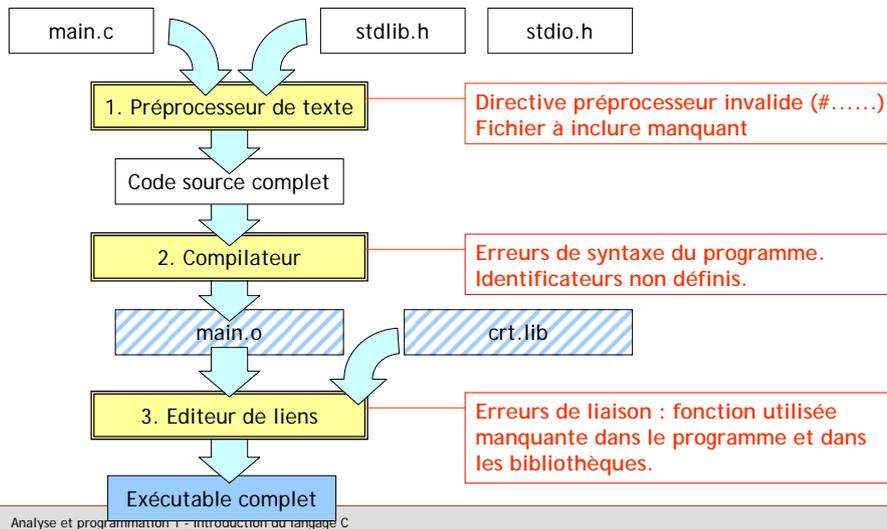
Déroulement en 3 phases



heig-vd

La compilation, un processus complexe

Les erreurs pouvant survenir



heig-vd

Différents types de programmes

Pouvant tous être créés avec le langage C

- Avec interface utilisateur
 - Les applications graphiques
 - Toutes les applications Windows : bloc note, navigateur Web
 - Les applications en mode console
 - L'utilisateur tape une commande pour donner un ordre.
 - Très utilisé lorsqu'un ordinateur ne gère que le mode caractère.
- Sans interface utilisateur
 - Services
 - Programmes non visibles de Windows.
 - Exemple : gestion du réseau.
 - Informatique embarquée
 - Programmes chargés dans des microcontrôleurs.

Différents types de programmes

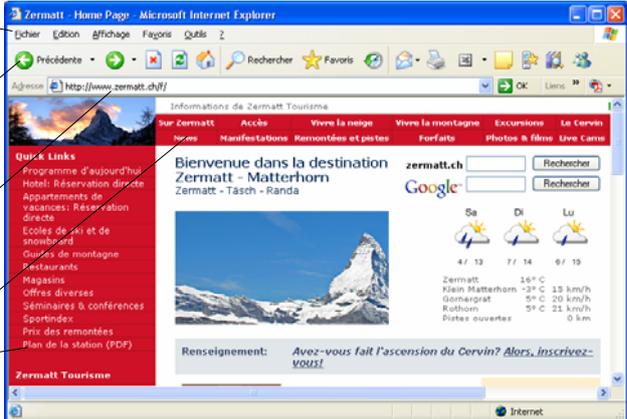
Les applications graphiques

Gérée avec des menus

Et des barres d'outil

L'utilisateur peut saisir du texte

et aussi cliquer un peu partout, tout ça en même temps



Différents types de programmes

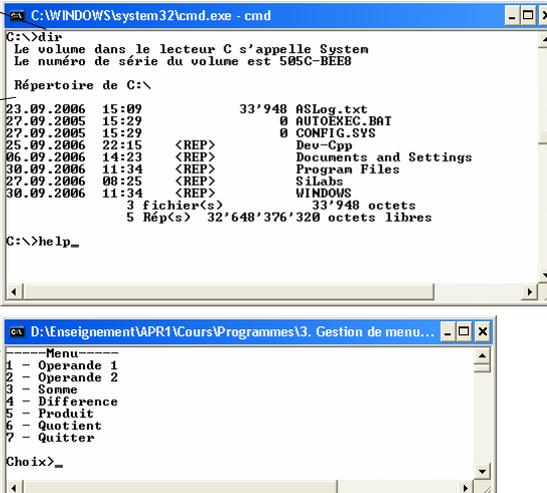
Les applications en mode console

Le programme «cmd.exe» permet de taper des commandes DOS

Et de voir le résultat

Une application console peut aussi guider l'utilisateur avec un menu.

On peut les basculer en plein écran (alt + Entrée)



heig-vd

Différents types de programmes

Les applications embarquées - Pour quoi faire ?

- Elles gèrent des entrées / sorties physiques.
 - Par exemple sur des machines automatisées



heig-vd

Différents types de programmes

Les applications embarquées - Comment les créer ?

- Elles sont développées avec des outils dédiés.



heig-vd

Différents types de programmes

Quel type d'applications allons nous créer ?

- Durant les travaux de laboratoire
 - Nous créerons des applications « Console » (ligne de commande)
- Motivations
 - Ce type d'applications est le plus facile à créer.
 - Il est universel, il existe dans tous les environnements.
 - Même possible sur une majorité de micro contrôleurs.
 - Exige moins de connaissance pour pouvoir commencer.
 - Permet de se focaliser sur l'apprentissage des fondements de l'analyse et de la programmation.
 - Les principes appris restent valables pour tous les types d'application.

heig-vd

Comment exécuter un programme compilé ?

Démarrage d'une application

- Avec un système d'exploitation
 - Tous les systèmes d'exploitation offrent des moyens pour démarrer un programme
 - Par exemple, sous Windows, il suffit de double cliquer sur le fichier exécutable.
 - Sous DOS, il faut taper le nom du programme.
- Sur un microcontrôleur (informatique embarquée)
 - L'outil de développement spécifique utilisé offre des fonctions pour
 - Envoyer le programme dans la mémoire du micro contrôleur.
 - Démarrer son exécution.

heig-vd

Les outils de développement

Un vaste choix

- Le langage C est largement utilisé dans l'industrie.
 - Un très grand nombre d'outils sont disponibles.
- On distingue 2 familles principales
 - Les outils en ligne de commande
 - Le code source doit être créé avec un éditeur de texte séparé.
 - Il doit ensuite être enregistré dans un fichier.
 - Ce fichier est compilé en tapant une ligne de commande
 - `cc -o programme1 main.c`
 - Les environnements de développement intégrés (IDE)
 - L'éditeur, le compilateur et d'autres outils sont intégrés.
 - Ils sont accessibles très facilement dans les menus d'un environnement graphique.
- Certains outils sont chers, d'autres sont gratuits.

heig-vd

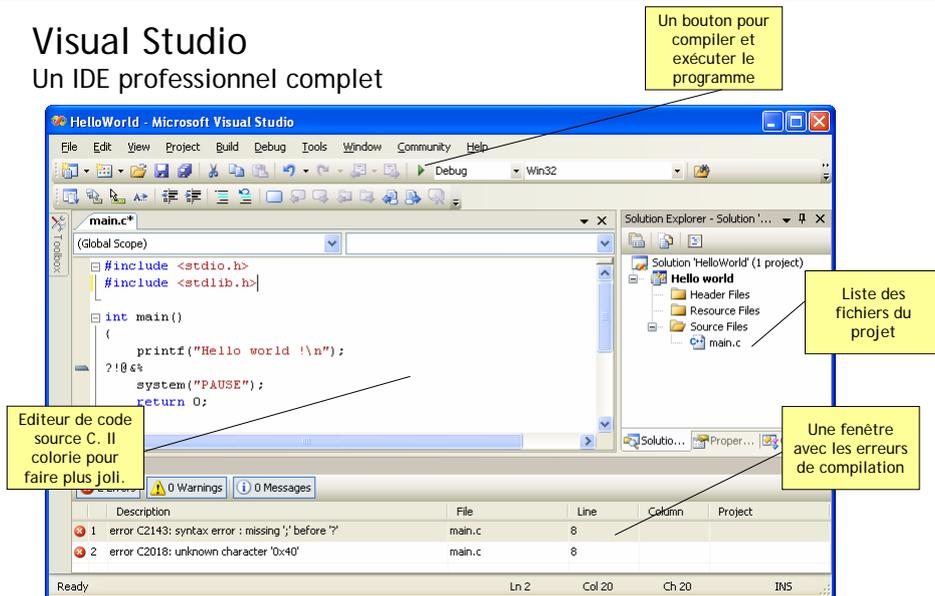
Les outils de développement

Lequel choisir

- Recherche sur internet
 - Des dizaines d'outils de programmation pour le C gratuits.
 - Des centaines d'outils payants.
 - Beaucoup d'outils sans IDE.
- Outils très connus et gratuits
 - GNU CC : compilateurs open source. Intégré dans de nombreux IDE.
 - Visual Studio C++: environnement professionnel très complet.
- **Choix retenu**
 - Microsoft Visual C++ 2005

Visual Studio

Un IDE professionnel complet



Petite histoire du langage C

De l'apparition à la normalisation

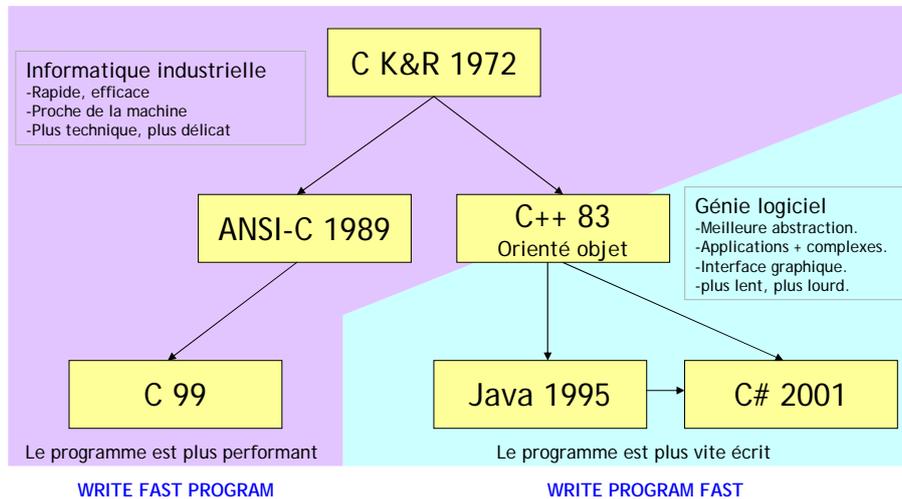
- 1950 : Fortran -> 1958 Algol -> 1963 CPL -> 1966 BCPL
- 1969 : Langage B, BCPL optimisé par Ken Thompson.
- 1972 : Langage C, par Brian Kernighan et Dennis Ritchie.
- 1983 : Extension orientée objet C++, par Bjarne Stroustrup
- 1989 : Normalisation ANSI-C, appelée C89
- 1990 : Normalisation ISO, identique à C89, appelée C90
- 1999 : Nouvelle normalisation ISO, appelée C99
- 2005 : Révision de la norme C99

• Pour en savoir plus

- http://en.wikipedia.org/wiki/C_programming_language#ANSI_C_and_ISO_C

Petite histoire du langage C

Les descendants



Petite histoire du langage C

Très largement répandu aujourd'hui

- Pour toutes les applications embarquées
 - La majorité des microcontrôleurs ont un compilateur C.
 - Le C++ est aussi disponible pour les microcontrôleurs les plus performants.
- Dans le C++
 - Le C++ est une extension du langage C.
 - Les principales notions acquises restent valables.
 - Un grand nombre d'applications PC actuelles sont écrites en C++.
- Dans le langage Java et C#
 - Ces langages s'appuient pour une grande part sur la syntaxe C.
 - La transition de C vers Java ou C# est donc facilitée.

Les faiblesses du langage C

Il n'en manque pas...

- C'est un langage ancien
 - Pensé à une époque où les applications étaient toutes petites.
 - Premier système Unix : fonctionnait avec 8 ko de RAM !
 - Syntaxe prévue pour permettre une compilation facile.
 - Grande liberté : autorise des constructions bizarres
 - Très efficaces à l'exécution.
 - Difficiles à lire et à comprendre pour le programmeur.
 - Sources de nombreuses erreurs.
 - Détecte peu d'erreurs de construction de programme
 - Contre partie de la liberté offerte.
- Limites d'utilisation
 - Pour les petites applications
 - Notamment lorsque l'efficacité est critique
 - Pour les applications plus complexes, aller vers C++.

Pourquoi apprendre le langage C aujourd'hui

Motivations du choix

- Valeur ajoutée pour les étudiants
 - Le langage C est très largement utilisé dans l'industrie.
 - Utilisé dans les laboratoires de plusieurs filières à la HEIG.
- Vertus pédagogiques
 - Amène une compréhension approfondie du fonctionnement d'un ordinateur et d'un programme.
 - Les acquis restent valables pour C++, Java, C#
 - Il est plus facile de passer de C à C++, Java, ou C# que l'inverse.
- Limitations
 - Les compilateurs C imposent peu de rigueur
 - Il en faut cependant beaucoup pour atteindre un résultat fonctionnel.
 - Demande plus d'efforts de compréhension aux étudiants.
 - La correction des erreurs exige souvent la compréhension de ce qui se passe à bas niveau dans l'ordinateur.

heig-vd

Qu'avons nous appris ?

- Le langage C
 - Est un langage textuel.
 - Facilite la programmation par comparaison au langage machine.
 - Permet de faire des programmes très performants.
 - Est très utilisé dans l'informatique embarquée.
 - Doit être compilé en langage machine pour être exécutable.
 - Les IDE facilitent grandement le cycle « édition, compilation, test »
 - Base syntaxique du C++, Java, et C#.
 - Chaque langage a un domaine d'utilisation de prédilection.

heig-vd

Vos questions

heig-vd

