

Langage C

- Langage impératif
- 1978 Parution du "*Kernighan et Richie*"
- 1989 Normalisation C-Ansi
- Langage compilé : source, code objet, exécutable.

Contexte

- Langage dans lequel sont écrits les SE Unix.
- Réputé pour ses performances - accès très bas niveau au matériel.
- Permet la gestion de périphériques : laser, microscope, spectro ...
- Nombreuses bibliothèques standards : mathématiques, entrée/sortie standard ...
- Utilisation des modules GNU : gcc, make, gdb

Mon premier programme

```
# include <stdio.h>
int main(int argc, char *argv[])
{
    printf("hello world \n");
}
```

Syntaxe

- Les commentaires : `//` une ligne ou `/*` plusieurs lignes `*/`
- Toutes instructions se termine par un `;`
- Déclaration des variables : `type nomVariable;`
- Les types :
 - `int`, `double`, `float`
 - pas de booléens
 - `char` //pas de type chaine
 - `[]` et `struct`
 - les constantes :

```
#define MAX 10;
```
- Absence de type `void`.

Syntaxe

- Déclarer une fonction :

```
typeRetour nomFonction( typeparam1, typeparam2);
```

- Définir une fonction:

```
typeRetour nomFonction(type1 param1 , type2 param2)  
{  
    instructions  
}
```

- Une fonction dont le type de retour est `void` n'a pas de valeur de retour.

Structures de contrôle

- Test if :

```
if (condition)
  { instructions;}
else
  { instructions;}
```

- Test switch :

```
switch (rep)
{
  case 1 : instructions;break;
  case 2 : instructions;
  default : instructions;
}
```

Structures de contrôle

- Boucle for :

```
for (i=0; i<10;i++)  
    {print(i);}
```

- Boucle while :

```
int i=0;  
while (i<10)  
{  
    print (i);  
    i++;  
}
```

- Boucle dowhile

Opérateurs

- Arithmétique :
 - + ++ - -- * /
- Comparaison :
 - < > <= >= == !=
- Logique :
 - ||, &&, !
- Opérations sur les bits :
 - << >> | &

Les Tableaux

```
#include <stdio.h>
#define MAX_LEN 20
int main(int argv, char * argc)
{
    int val[MAX_LEN];
    int i;
    for (i=0;i<MAX_LEN;i++)
    {
        val[i]=1;
    }
    for (i=0;i<MAX_LEN;i++)
    {
        printf("%d \n",val[i]);
    }
}
```

Les entrées - sorties

```
#include <stdio.h>
int main(int argv, char * argc)
{
    char c;
    while ((c=getchar()) !=EOF)
    {
        putchar(c);
    }
}
```

Les entrées - sorties

```
#include <stdio.h>
int main(int argv, char * argc)
{
    char chaine[Max_LEN+1];
    while (fgets(chaine,MAX_LEN+1,stdin) !=NULL)
    {
        puts(chaine);
    }
}
```

Les entrées-sorties

- Fichiers de texte formatés ou non.
- Accès séquentiels.
- Gestion des fichiers : `open`, `fopen`, `fread`, `fwrite`, `fclose`, `fseek`, `feof`, `fflush` et toutes les fonctions d'entrée/sortie standard préfixées par `f`
- Le type de retour est `File *`.

Exemple de code

```
File *fic=fopen(`toto.txt`,`r`);  
while (fscanf(fic,`%s`,chaine) !=EOF)  
    printf(`%s`,chaine);  
fclose(fic);
```