

Langage C/C++

Cours 1/5 : bases du langage

Hubert Godfroy

13 novembre 2014

Plan

Intérêts du C/C++

Éléments de base

Prise en main de l'IDE

Plan

Intérêts du C/C++

Éléments de base

Prise en main de l'IDE

Histoire du C

- ▶ 1978 : invention par Kernighan et Ritchie
- ▶ 1984 : spécification par l'ANSI (American National Standards Institute)
- ▶ Une extension orientée objet avec le C++
- ▶ Énormément de nouveaux langages sont dérivés du C

Compréhension

Les systèmes sont écrit en majorité en C.

- ▶ Langage de bas niveau
- ▶ Il est possible de programmer n'importe quoi.
- ▶ Accès direct à la mémoire.
- ▶ Pas de garde-fou (un grand pouvoir implique de grandes responsabilités).

Facilite la compréhension des autres langages

- ▶ Explicite les mécanismes en jeu dans les autres langages

Plan

Intérêts du C/C++

Éléments de base

Prise en main de l'IDE

Comparaison avec JAVA

- ▶ Utilisation des mêmes opérateurs (+, -, *, /, %, ++, &&, ||, ==, ...)
- ▶ Même structures de contrôle

Exemple

```
for(i = 0; i < 10; i ++){...}  
    while(i < 10){...}  
        if(i < 10){...}
```

Comparaison avec JAVA

- ▶ Utilisation des mêmes opérateurs (+, -, *, /, %, ++, &&, ||, ==, ...)
- ▶ Même structures de contrôle

Exemple

```
for(i = 0; i < 10; i ++){...}  
    while(i < 10){...}  
        if(i < 10){...}
```

Mais :

- ▶ pas de GC (gestion manuelle de la mémoire)
- ▶ pas de packages

Variables

Processus en deux temps :

Déclaration

```
int i
```

À partir de ce moment, la variable à cours ***dans le bloc courant***

Définition

```
i = 42
```

Variable

Attention à la **portée** des variables

- ▶ Un bloc correspond à un ensemble de code entre deux accolades correspondantes.
- ▶ Une variable a cours dans toutes les instructions du bloc *après* sa déclaration.
- ▶ La variable est détruite à la sortie du bloc l'ayant déclarée.

(cf exercices 1-3 du TD 1)

Types de base

- ▶ Une partie des types standards habituels : `char`, `int`, `float`.
- ▶ Absence de types usuels : `boolean`, `string`.
- ▶ Nouveaux types : `unsigned char`, `unsigned int`.
- ▶ D'autres types structurés (cf prochain cours)

Entrée/sorties

Se font avec `scanf` et `printf` :

Entrée/sorties

Se font avec `scanf` et `printf` :

Exemple de sortie

```
printf("Hello world")  
printf("La réponse est %d", 42)  
printf(" $\pi$  = %f", 3.14)
```

Entrée/sorties

Se font avec `scanf` et `printf` :

Exemple de sortie

```
printf("Hello world")  
printf("La réponse est %d", 42)  
printf(" $\pi$  = %f", 3.14)
```

Exemple d'entrée

```
scanf("%d", &taille)  
scanf("%c", &lettre)
```

Déclaration de fonctions

Syntaxe

```
type0 nom_fonction (type1 x1, ...){  
    ...  
}
```

Déclaration de fonctions

Syntaxe

```
type0 nom_fonction (type1 x1, ...){  
    ...  
}
```

Exemple

```
int irma (int n){return (rand() % n)}
```

Déclaration de fonctions

Syntaxe

```
type0 nom_fonction (type1 x1, ...){  
    ...  
}
```

Exemple

```
int irma (int n){return (rand() % n) }
```

Remarque

On peut aussi séparer déclaration et définition d'une fonction. Ici la déclaration de `irma` est

```
int irma (int)
```

Exemple minimal

```
#include <stdio.h>
int double(int n){
    return n*2;
}
int main(int argc, char *argv[]){
    return double(5);
}
```

Plan

Intérêts du C/C++

Éléments de base

Prise en main de l'IDE

- ▶ Environnement de travail Qt
- ▶ Environnement de développement intégré
- ▶ Aide à la programmation : auto-complétion sémantique, réusinage, aide à la correction d'erreurs, ...

DEMO

Et maintenant, à vous de jouer !