

prendre des arguments en ligne de commande

christophe.labourdette(at)cmla.ens-cachan.fr

Octobre 2016

déclaration du main

```
#include <string>
#include <iostream>
int main (int argc, char* argv[])
{
    string mot;
    for (int i=0;i<argc;i++)
    {
        mot = argv[i];
        std::cout << mot << "\n";
    }
    std::cout <<std::endl;
}
```

exécution

L'exécution, après compilation du programme précédent donne par exemple ceci :

```
./exmain on peut le dire avec des fleurs ...  
./exmain  
on  
peut  
le  
dire  
avec  
des  
fleurs  
...
```

Quelques petites remarques :

- argc contient le nombre d'arguments
- le premier argument est le nom du programme
- les arguments sont séparés par des blancs
- les arguments sont des chaînes de caractères
- si l'on veut récupérer autre chose il faut convertir

Récupérer un entier ou un flottant

On doit utiliser une conversion pour transformer une string en entier ou flottant :

- `to_string (val)` convertie une valeur en string ,
- `stoi (str)` convertie une string en int ,
- `stof (str)` convertie une string en float ,
- `stod (str)` convertie une string en double.

exemple

```
#include <string>
#include <iostream>
int main (int argc, char* argv[])
{
    std::string mot;
    double a, b;
    mot = argv[1];
    a = std::stod(mot);
    mot = argv[2];
    b = std::stod(mot);
    std::cout << a << "+" << b << "=";
    std::cout << a+b<<"\n";
}
```

exécution

l'exécution du programme précédent, après compilation
(`g++ exmain2.cpp -o exmain2 -std=c++11`)
donne le résultat suivant :

```
./exmain2 67.97e+2 88.45  
6797+88.45=6885.45
```

Les deux chaînes de caractères ont bien été transformées en double, ce qui nous a permis de les additionner