

JI 2014 : Atelier D

Christophe Meessen

- Installation du compilateur D

<http://dlang.org/download.html>

- Windows :

- utiliser powershell.exe
- Allez dans le répertoire de travail
 - > cd \users\<nom>\...
- Activez le support des caractères utf8
 - > chcp 65001



- Compilation et exécution du programme Hello World !

Fichier Hello.d :

```
import std.stdio;  
void main() {  
    writeln("Hello world!");  
}
```

Windows :

> dmd Hello.d (compilation)

> Hello.exe (exécution)

Hello world !

> rdmd Hello.d (compilation et exécution combinée)

Hello world !

Linux :

> dmd Hello.d (compilation)

> ./Hello (exécution)

Hello world !

> rdmd Hello.d (compilation et exécution combinée)

Hello world !

Ajouter `#!/usr/bin/rdmd` au début du fichier et rendre Hello.d exécutable : > `chmod u+x Hello.d`

> ./Hello.d (compilation et exécution combinée)

Hello world !

- Programme affichant les arguments

Fichier exo1.d:

```
import std.stdio;
void main(string[] args) {
    writeln("Arguments :");
    for (int i = 0; i < args.length; ++i)
        writefln("%3s : %s", i, args[i]);
}
```

> rdmd exo1.d Donald Duck

Arguments :

0 : /tmp/.rdmd-5175/rdmd-exo1.d-F549817.../exo1

1 : Donal

2 : Duck

- Programme utilisant les arguments

Fichier exo2.d:

```
import std.stdio;
void main(string[] args) {
    write("Hello");
    if (args.length < 2)
        writeln(" world!");
    else {
        for (int i = 1; i < args.length; ++i)
            writef(" %s", args[i]);
        writeln("!");
    }
}
```

> rdmd exo2.d Donald Duck

Hello Donald Duck !

- Programme utilisant les arguments

Fichier exo3.d:

```
import std.stdio, std.array;
void main(string[] args) {
    args.popFront;
    if (args.length == 0)
        args ~= "world";
    write("Hello");
    foreach (arg; args)
        writef(" %s", arg);
    writeln("!");
}
```

```
> rdmd exo3.d Donald Duck
Hello Donald Duck !
```

- Indexer un répertoire

Fichier exo4.d:

```
import std.stdio, std.file;
void main(string[] args) {
    string p = args.length>1 ? args[1] : ".";
    if(!p.exists)
        return;
    if(p.isDir)
        foreach(e; dirEntries(p, spanMode.breadth))
            writefln("%7s %s", e.size, e.name);
    else {
        auto e = DirEntry(p);
        writefln("%7s %s", e.size, e.name);
    }
}
```

- Compilation des programmes D en parallèle

Fichier exo5.d:

```
import std.stdio, std.file;
import std.algorithm, std.parallelism;
void main() {
    // backticks délimitent un string raw
    auto dFiles = filter! `endsWith(a.name, ".d") `
                    (dirEntries(".", SpanMode.depth));
    // passe un fichier D à chaque thread
    foreach (d; parallel(dFiles, 1))
    {
        string cmd = "dmd -c " ~ d.name;
        writeln(cmd);
        std.process.system(cmd);
    }
}
```


- Écriture dans un fichier texte

Fichier exo6.d:

```
import std.stdio, std.file, std.math;
void main() {
    immutable nVal = 40;
    immutable real pi = 3.14159265;
    immutable real dx = 2*pi/nVal;
    auto file = File("data.txt", "w+");

    real x = 0;
    for (int i = 0; i < nVal; ++i, x += dx )
        file.writeln("%s %s", x, sin(x) );
}
```

- Lecture d'un fichier texte et affichage du contenu

Fichier exo7.d:

```
import std.stdio, std.file;
void main() {
    auto file = File("data.txt", "r");
    real x, y;
    while (file.readf(" %s %s", &x, &y))
        writeln(x, " ", y);
}
```

- Écriture dans un fichier binaire

Fichier exo8.d:

```
import std.stream;
align(1) struct Item { align(1):
    char set = 'N';
    int id;
    float val;
}
void main() {
    Item[] idx = [1:{'Y', 10, 0.5},
                  {val:0.352, id:48, set:'Y'}];
    auto file = new File("f.idx", FileMode.Out);
    file.writeExact(&idx[0],
                   Item.sizeof*idx.length);
}
```

- Lecture d'un fichier binaire (1/2)

Fichier exo9.d:

```
import std.stream;
import std.stdio : writefln;

align(1) struct Item { align(1):
    char set = 'N';
    int id;
    float val;
}

void main() {
    Item[] idx;
    auto file = new File("f.idx", FileMode.In);
```

- Lecture d'un fichier binaire (2/2)

Fichier exo9.d (suite):

```
Item buffer;
while (!file.eof)
{
    file.readExact(&buffer, Item.sizeof);
    idx ~= buffer;
}
foreach (i; idx)
    writefln("%s %s %s", i.set, i.id, i.val);
}
```