



Version du logiciel : 1.3

Version du support : 1.0.1

Date de création : 2 novembre 2018

Date de dernière mise à jour : 1 mars 2020

IDE utilisées :

- Notepad++ pour les « applications » (.kt).
- IntelliJ de JetBrains pour les projets Kotlin avec des scripts (.kts).

Table des matières

1.1 - Présentation de ce document.....	3
1.1.1 - Objectif.....	3
1.1.2 - Bibliographie et Webographie.....	4
1.2 - Création d'un fichier ASCII.....	5
1.3 - Lecture d'un fichier ASCII.....	6
1.3.1 - Avec readText().....	6
1.3.2 - Avec File().forEachLine.....	6
1.3.3 - Avec readLines().....	7
1.3.4 - Avec useLines.....	8
1.4 - Ajout dans un fichier ASCII.....	9
1.5 - Autres techniques de lecture.....	10
1.5.1 - Avec InputStream.....	10
1.5.2 - Avec BufferedReader.....	11
1.6 - Autres techniques d'écriture.....	12
1.6.1 - Avec PrintWriter() et println().....	12
1.6.2 - Avec bufferedWriter() et write().....	12

1.1 - Présentation de ce document

1.1.1 - Objectif

Introduire à la bibliothèque IO.

Chemin :

C:\pascal__supports\kotlin\scripts_io\

1.1.2 - Bibliographie et Webographie

Bibliographie

<https://kotlinlang.org/docs/books.html>

En français :

Un seul livre (ENI éditions) ; ce livre est pour les développeurs Android mais comporte des chapitres sur kotlin « pur et dur ».

« Kotlin, les fondamentaux du développement d'applications Android »

Éditeur(s)	Eni
Auteur(s)	Anthony Cosson
Collection	Epsilon
Parution	10/10/2018
Nb. de pages	466
EAN13	9782409015861
ISBN13	978-2-409-01586-1
Prix	54 €

Webographie

Tutoriel officiel :

<https://kotlinlang.org/docs/tutorials/kotlin-for-py/file-io.html>

Référence officielle :

<https://kotlinlang.org/docs/reference/>

1.2 - Création d'un fichier ASCII

```
val f = File("path")
```

```
f.writeText("content")
```

```
// write_ascii_file.kt
fun main(args: Array<String>) {
    val f = java.io.File("1.txt")
    val content = "Hello world!\nAvec kotlin !!!\n"
    f.writeText(content)
    println("Writed to file")
}
```

1.3 - Lecture d'un fichier ASCII

1.3.1 - Avec readText()

```
// read_ascii_file_others.kt
import java.io.File
fun main(args: Array<String>) {
    // with readText
    val content = File("1.txt").readText()
    println(content)
}
```

1.3.2 - Avec File().forEachLine

Le contenu du fichier « data.txt » stocké dans le même dossier que le script.

**Il fait très beau aujourd'hui ...
Mais pour demain ?**

Ouvre le fichier.
Boucle sur chaque ligne du fichier.
Ferme le fichier.

```
java.io.File().forEachLine { println(it) }
```

```
// read_ascii_file.kt
fun main(args: Array<String>) {
    java.io.File("data.txt").forEachLine { println(it) }
}
```

Affiche :

Il fait très beau aujourd'hui ...
Mais pour demain ?

1.3.3 - Avec `readLines()`

```
// read_ascii_file_others.kt

import java.io.File

fun main(args: Array<String>) {
    // with readLines
    val lines: List<String> = File("1.txt").readLines()
    lines.forEach { line -> println(line) }
}
```

Ceci est aussi possible !

```
val lines = File("1.txt").readLines()
```

Et en une seule ligne !

```
File("1.txt").readLines().forEach { line -> println(line) }
```

1.3.4 - Avec useLines

```
// read_ascii_file_others.kt

import java.io.File

fun main(args: Array<String>) {
    // with useLines
    val list = mutableListOf<String>()
    File("1.txt").useLines { lines -> list.addAll(lines) }
    list.forEachIndexed { i, line -> println("${i}: " + line) }
}
```

Affiche :

0: Un

1: Deux

1.4 - Ajout dans un fichier ASCII

```
Files.write(f.toPath(), content.toByteArray(), StandardOpenOption.APPEND)
```

```
// append_in_ascii_file.kt
package fr.pb // facultatif

import java.io.File
import java.nio.file.Files
import java.nio.file.StandardOpenOption

fun main(args: Array<String>) {
    val f = File("1.txt")
    val content = "Hello world!\nAvec kotlin !!!\n"
    Files.write(f.toPath(), content.toByteArray(), StandardOpenOption.APPEND)
    println("Append into file")
}
```

1.5 - Autres techniques de lecture

1.5.1 - Avec InputStream

Ouvre un Stream.

```
var ins: InputStream = f.inputStream()
```

Ferme un Stream.

```
ins.close()
```

```
// read_ascii_file_others.kt

import java.io.File
import java.io.InputStream
import java.nio.charset.Charset

fun main(args: Array<String>) {
    // with InputStream
    val f = File("1.txt")
    var ins: InputStream = f.inputStream()
    var content = ins.readBytes().toString(Charset.defaultCharset())
    ins.close()
    println(content)
}
```

Note :

is est un mot réservé ! D'où le choix de ins !

1.5.2 - Avec *BufferedReader*

I

1.6 - Autres techniques d'écriture

1.6.1 - Avec *PrintWriter()* et *println()*

```
// write_ascii_file_others.kt

import java.io.File

fun main(args: Array<String>) {
    val f = File("1.txt")
    // with printWriter
    f.printWriter().use { out ->
        out.println("1")
        out.println("2")
    }
    println("Writed to file")
}
```

1.6.2 - Avec *bufferedWriter()* et *write()*

```
// write_ascii_file_others.kt

import java.io.File

fun main(args: Array<String>) {
    val f = File("1.txt")
    // with bufferedWriter
    f.bufferedWriter().use { out ->
        out.write("Un\n")
        out.write("Deux\n")
    }
    println("Writed to file")
}
```